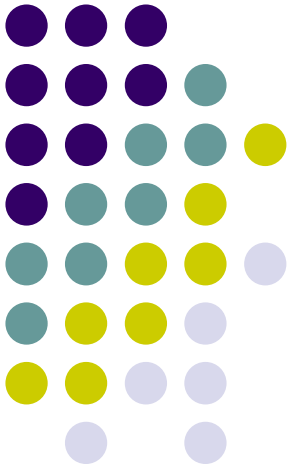
Introduction to NGS Analysis

Anna Battenhouse

abattenhouse@utexas.edu

June, 2023

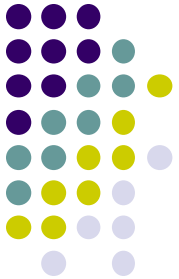


Associate Research Scientist

Center for Biomedical Research Support (CBRS)
Bioinformatics Consulting Group (BCG)
Biomedical Research Computing Facility (BRCF)
Genome Sequencing & Analysis Facility (GSAF)

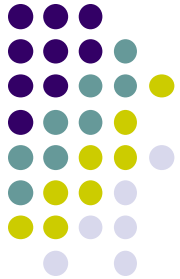
Center for Systems and Synthetic Biology (CSSB)
Ed Marcotte & Vishwanath Iyer labs

Goals

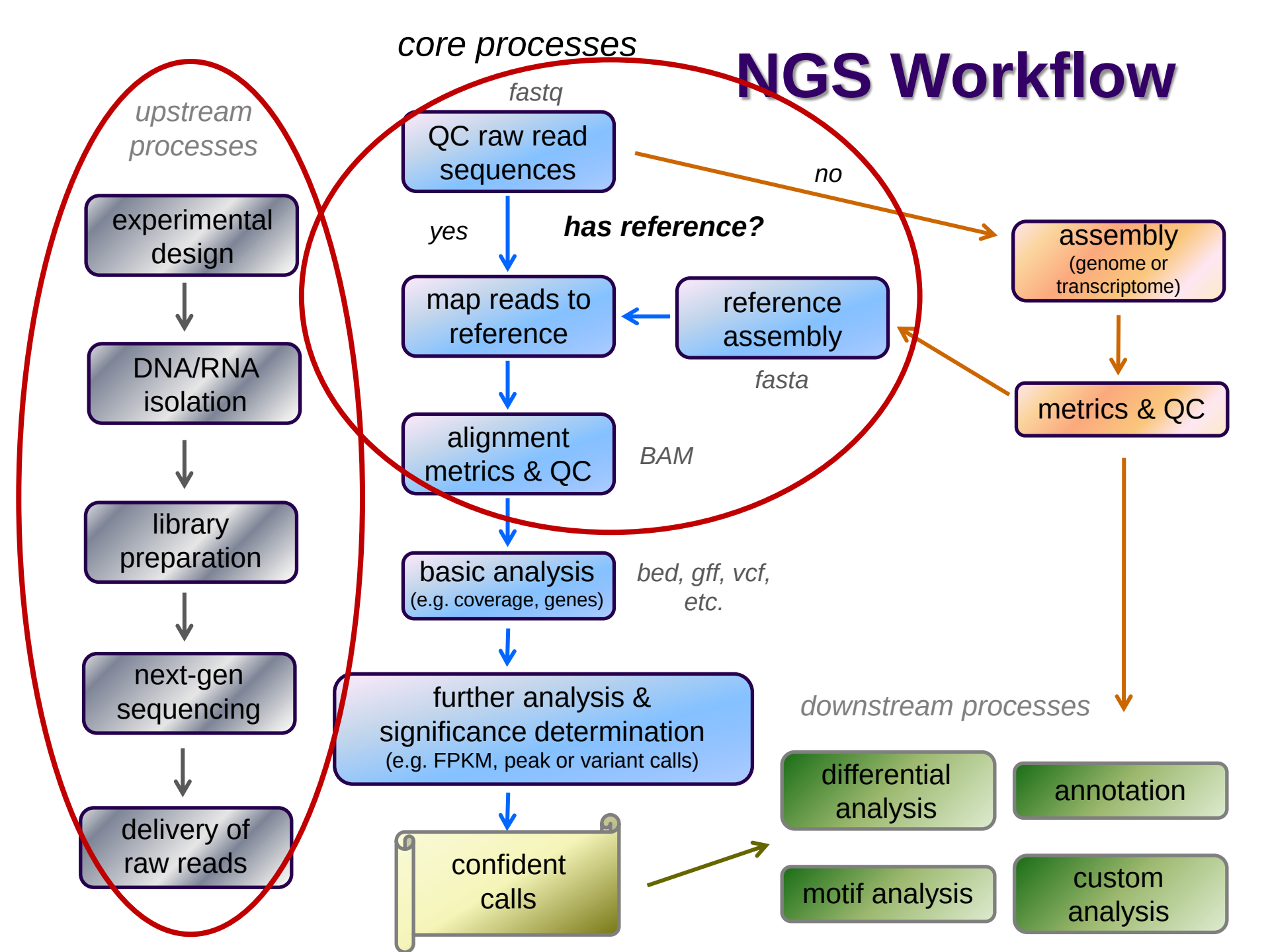


- Introduce NGS vocabulary
  - provide both high-level view and important consideration details
- Focus on common, initial tasks
  - raw sequence preparation, alignment to reference
  - common bioinformatics tools & file formats
- Understand required skills & resources
  - computational & storage resources
  - highlight best practices

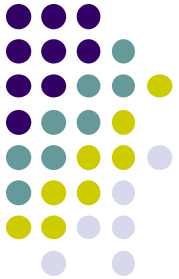
Other NGS Resources at UT



- CBRS short courses
  - 3-4 hour workshops on a variety of topics
    - Intro & Intermediate Unix; Advanced Bash scripting
    - Intro & Intermediate Python; Visualization in R
    - Intro to RNA-seq, single cell analysis
- Genome Sequencing & Analysis Facility (GSAF)
  - Jessica Podnar, Director, gsaf@utgsaf.org
- Bioinformatics consultants
  - Dennis Wylie, Dhivya Arasappan, Benni Goetz, Anna
  - Provide no-cost consulting on experimental design (with GSAF)
  - BioITeam wiki – https://wikis.utexas.edu/display/bioiteam/
- Biomedical Research Support Facility (BRCF)
  - provides local compute and managed storage resources
  - https://wikis.utexas.edu/display/RCTFUsers


core processes
NGS Workflow
upstream processes
fastq
QC raw read sequences
no
yes
has reference?
experimental design
map reads to reference
reference assembly
assembly
(genome or transcriptome)
fasta
DNA/RNA isolation
metrics & QC
library preparation
alignment metrics & QC
BAM
basic analysis
(e.g. coverage, genes)
bed, gff, vcf, etc.
next-gen sequencing
further analysis & significance determination
(e.g. FPKM, peak or variant calls)
downstream processes
differential analysis
annotation
delivery of raw reads
confident calls
motif analysis
custom analysis

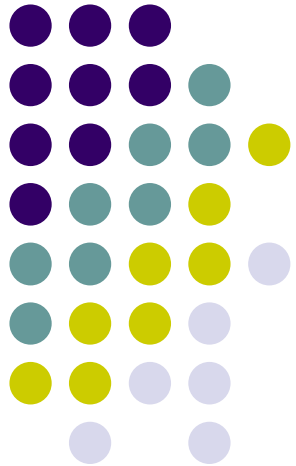
Outline



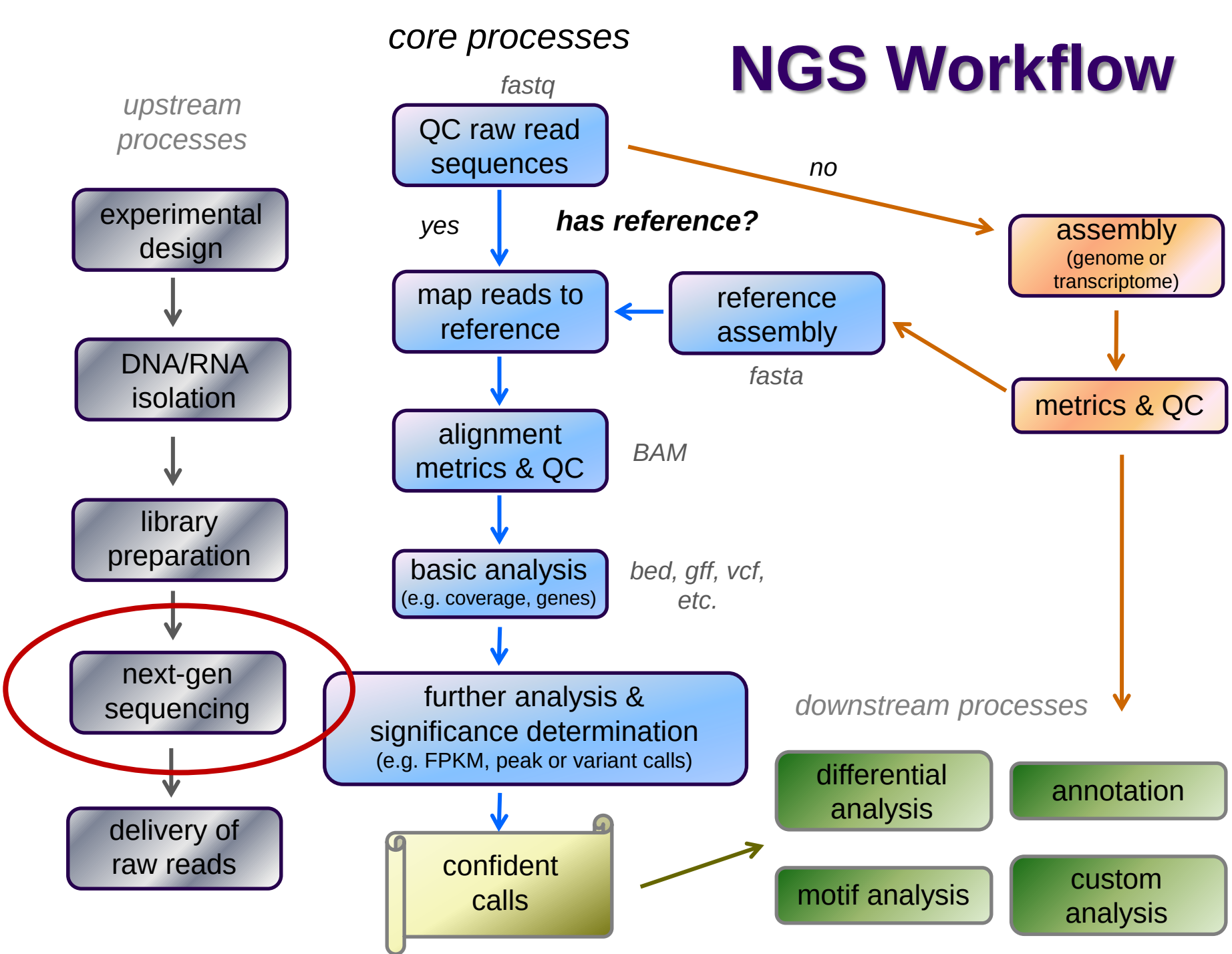
1. History of sequencing technologies
2. NGS terminology
3. The FASTQ format and
   Raw data QC & preparation
4. Alignment to a reference

Part 1: Overview of Sequencing Technologies

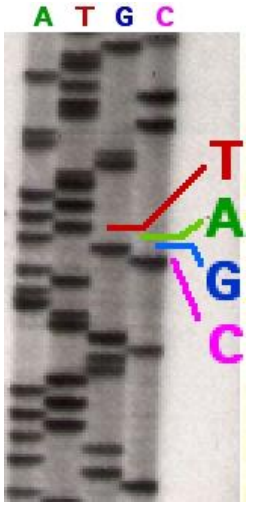
- Sanger sequencing
- The human genome project
- High-throughput (“next gen”) sequencing
- Illumina short-read sequencing
- Long read sequencing



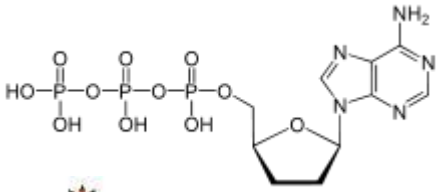
NGS Workflow

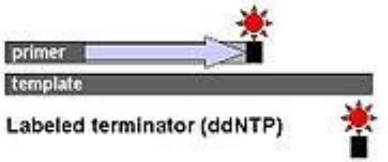


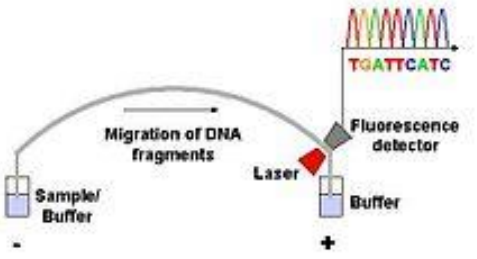
Sanger sequencing
(1st generation)

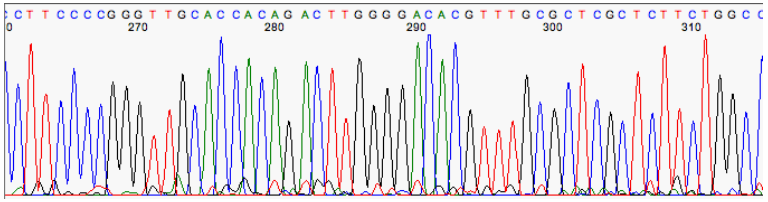


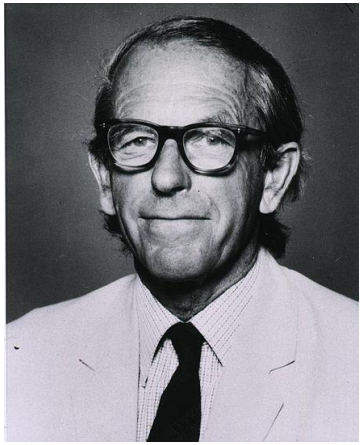
- Developed by Frederick Sanger, 1977
  - find sequence of one ***purified*** DNA molecular species
- Originally 4 sequencing reactions
  - all with deoxynucleotides (dNTs, e.g. dATP), DNA polymerase
  - each with different labeled chain-terminating ***ddNT***
    - ***dideoxynucleotide*** lacking 3'-OH
  - signal generated when ddNT incorporated
  - original signal from radiolabeling, readout on PAGE gel
- Now done in 1 reaction w/fluorescent dyes







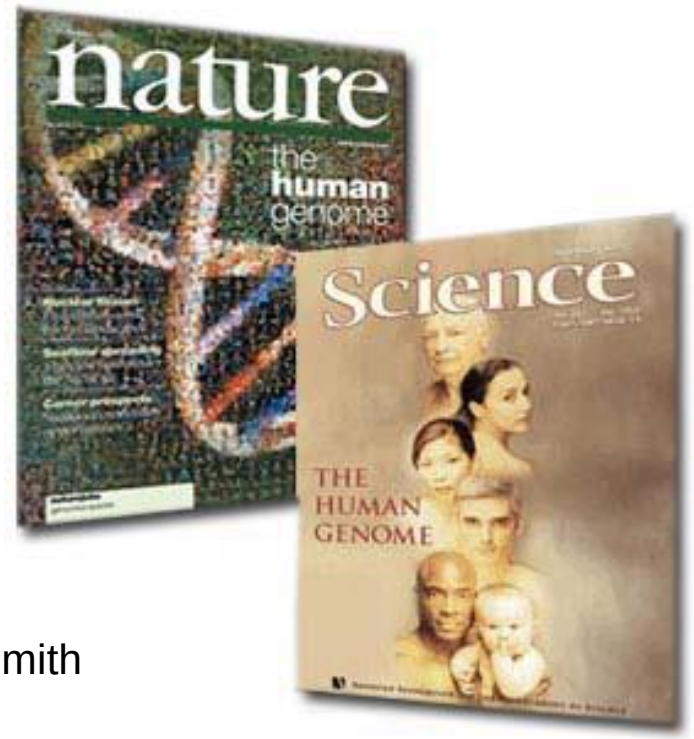




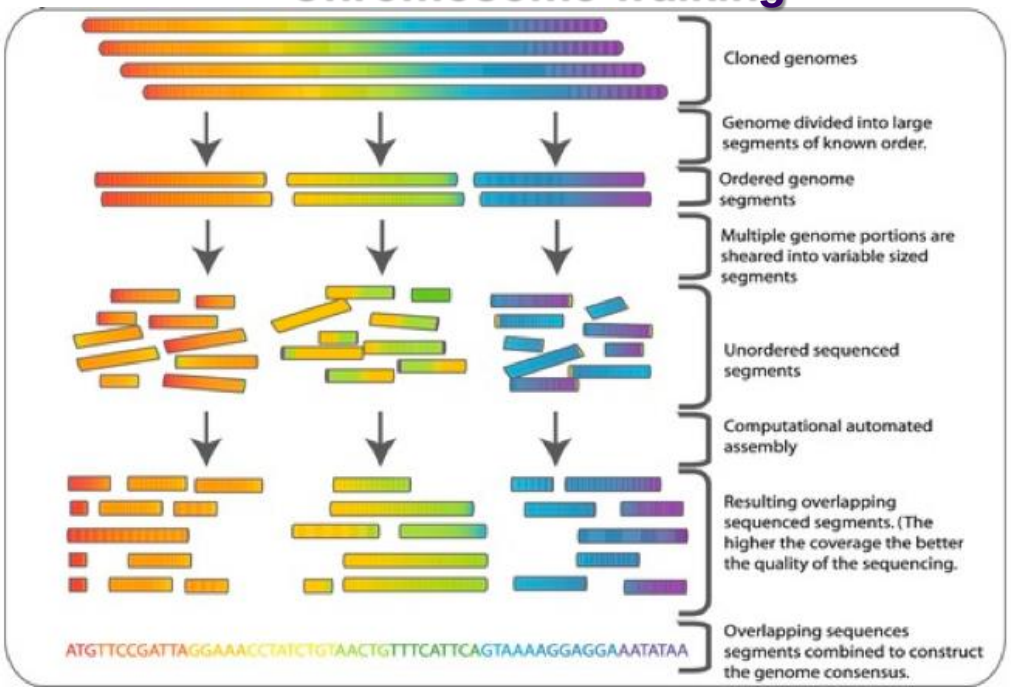
Frederick Sanger
1918 - 2013

Human Genome project

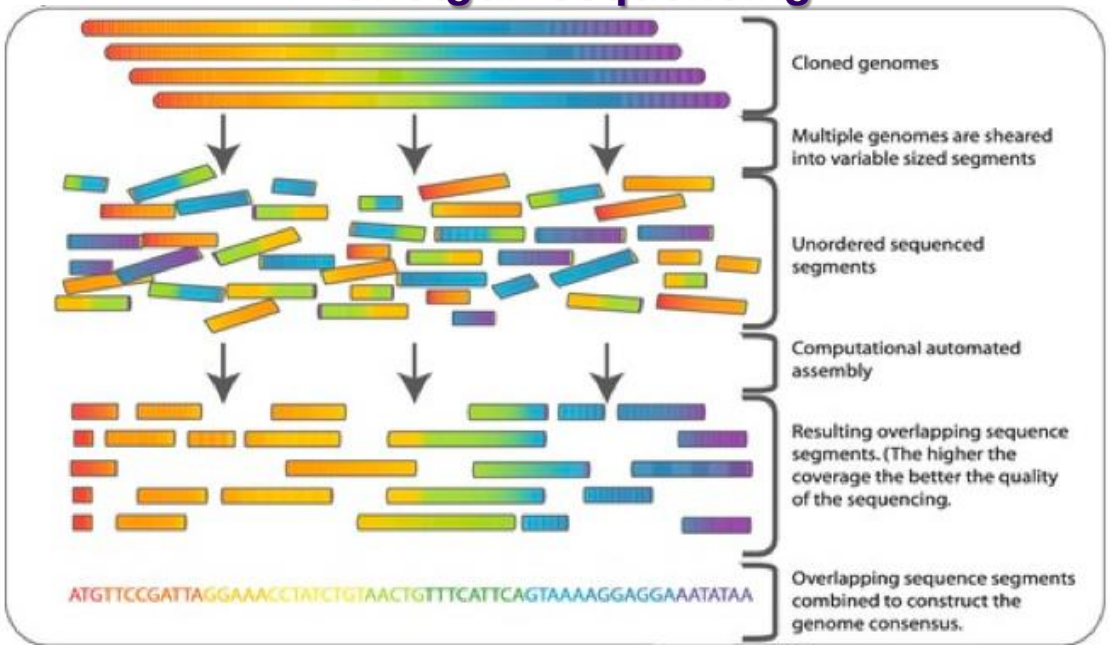
- Used Sanger sequencing to sequence ***3.3 billion*** base pair human genome!
- Massive effort
  - > 20 institutions worldwide
  - $2.7 billion cost
- Public effort started 1990
  - UCSC key player, Jim Kent
  - "***chromosome walking***" method
- Private effort started 1998
  - Celera Genomics, J. Craig Venter, Hamilton Smith
  - "***shotgun sequencing***" method
- 1st draft published jointly in 2001



Chromosome walking



Shotgun sequencing



Both

- Larger DNA fragments sheared into variable-sized segments
  - 2-50 kilobase (kb)
  - Sanger sequenced
- Fragments ***assembled*** computationally using partial overlaps
  - contiguous bases (***contigs***) placed onto larger ***scaffolds***
- High ***coverage*** (bases over a given position) required for reduced error ***consensus***

Chromosome walking

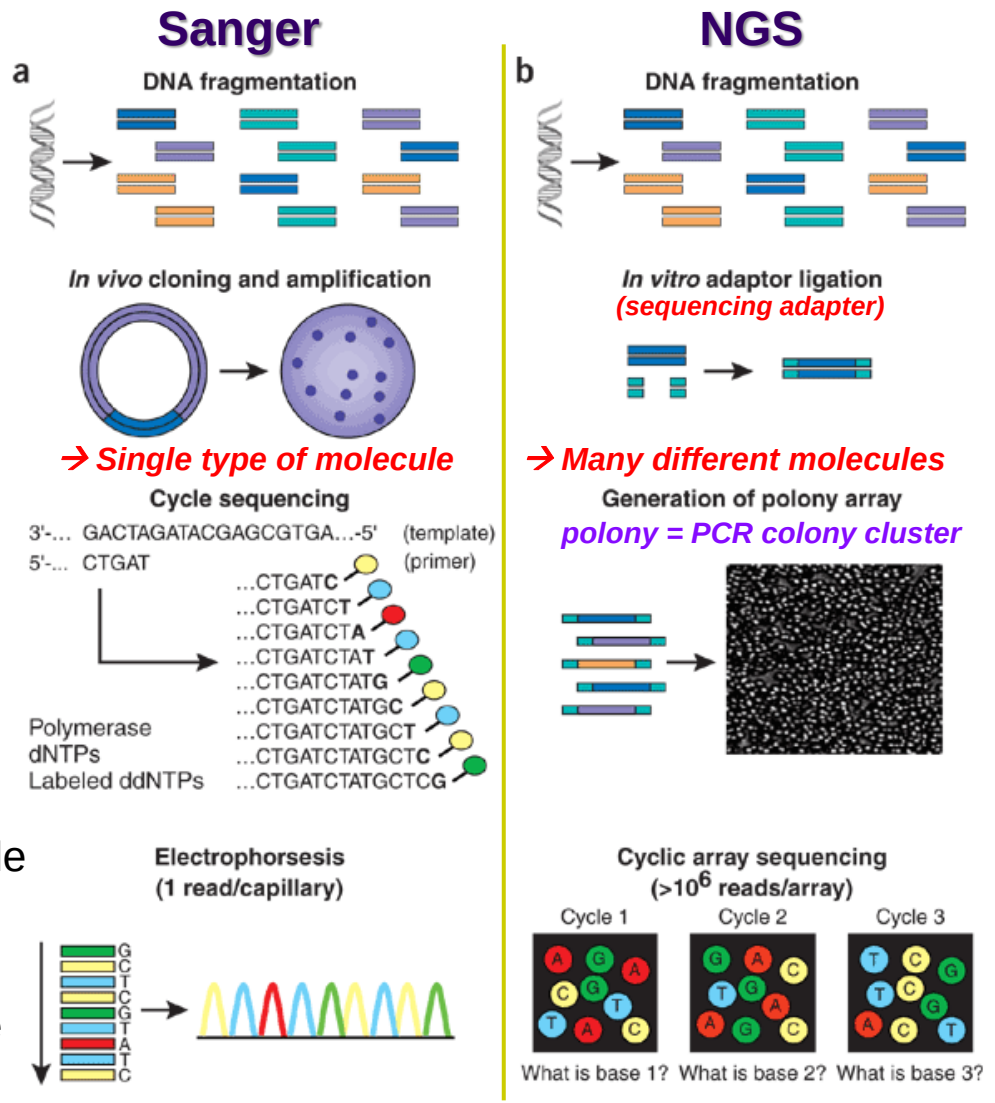
- 1st created large ***sub-clones*** with ***known order*** on genome

Shotgun sequencing

- Lack of large sub-clones made computational assembly more challenging

“Next Generation” sequencing

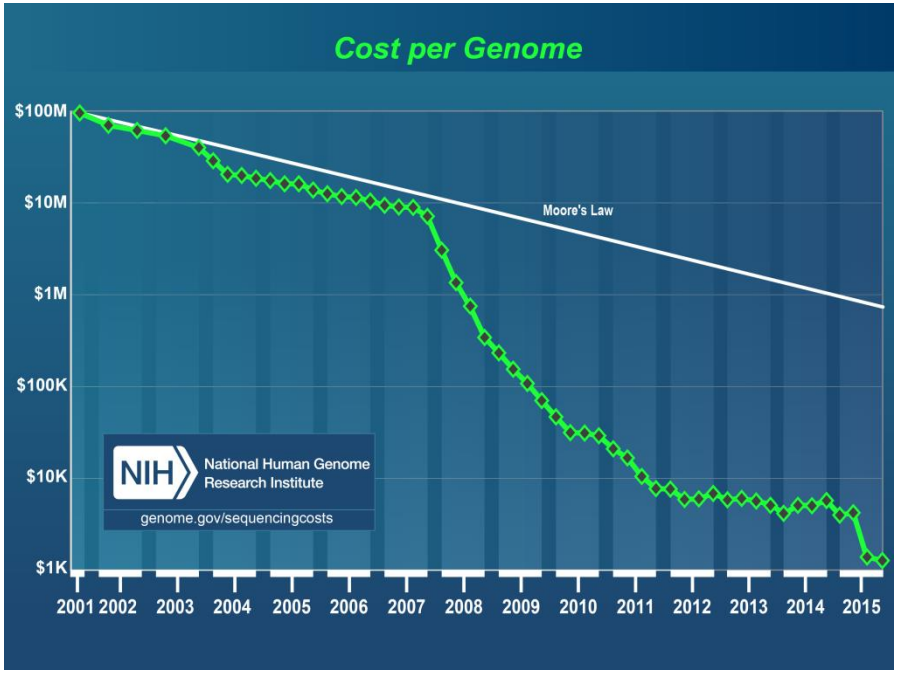
- Massively parallel
  - simultaneously sequence “***library***” of ***millions*** of ***different*** DNA fragments
- ***PCR colony clusters*** generated
  - individual template DNA fragments titrated onto a flowcell to achieve inter-fragment separation
  - PCR “bridge amplification” creates ***clusters*** of identical molecules
- ***Sequencing by synthesis***
  - fluorescently-labeled dNTs added
  - incorporation generates persistent signal (after wash)
  - flowcell image captured after each cycle
  - images computationally converted to base calls
    - including quality (confidence) measure
  - results in 30-300 base “reads”
    - *vs* multi-Kilobase with Sanger

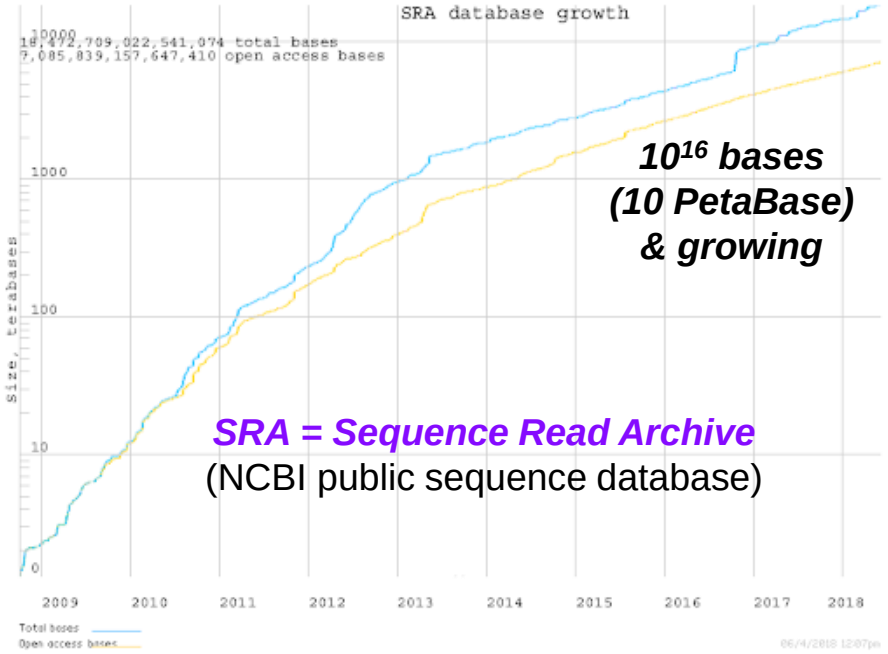


Shendure et al, Nature Biotechnology. 2008.
http://dx.doi.org/10.1038/nbt1486

“Next Generation” sequencing
(2nd generation)

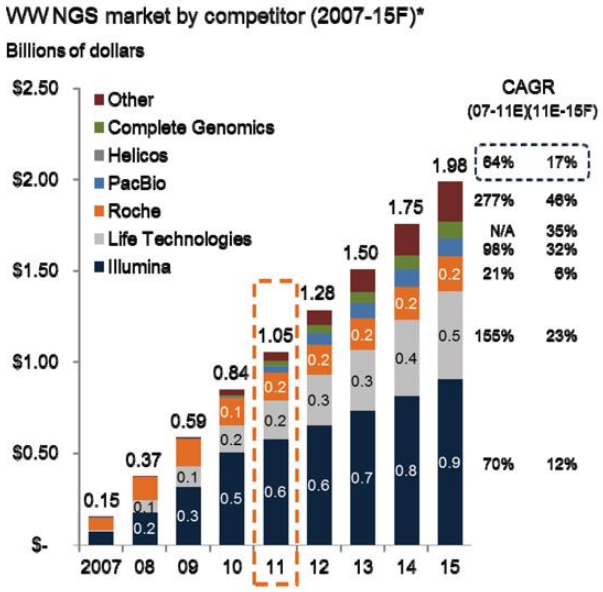
- Pro’s:
  - much faster!
  - much lower cost!
  - both deeper and wider coverage!
- Con’s:
  - data deluge!
  - storage requirements!
  - analysis lags!



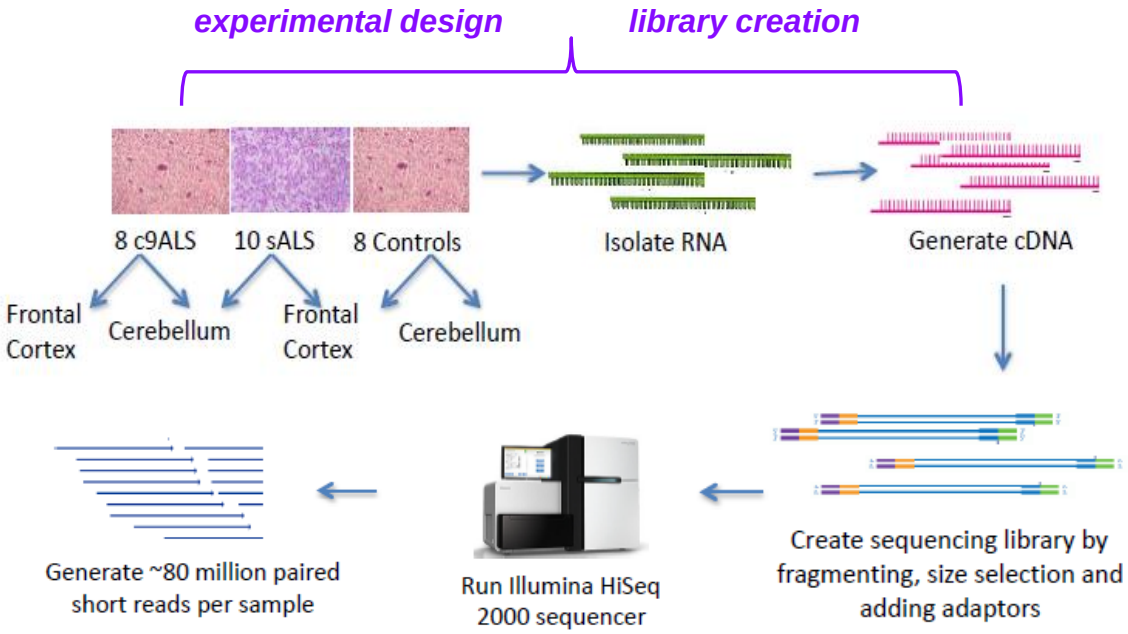


Illumina workflow

- Illumina dominant for “short” (<300 bp) reads



Typical Illumina RNA-seq workflow

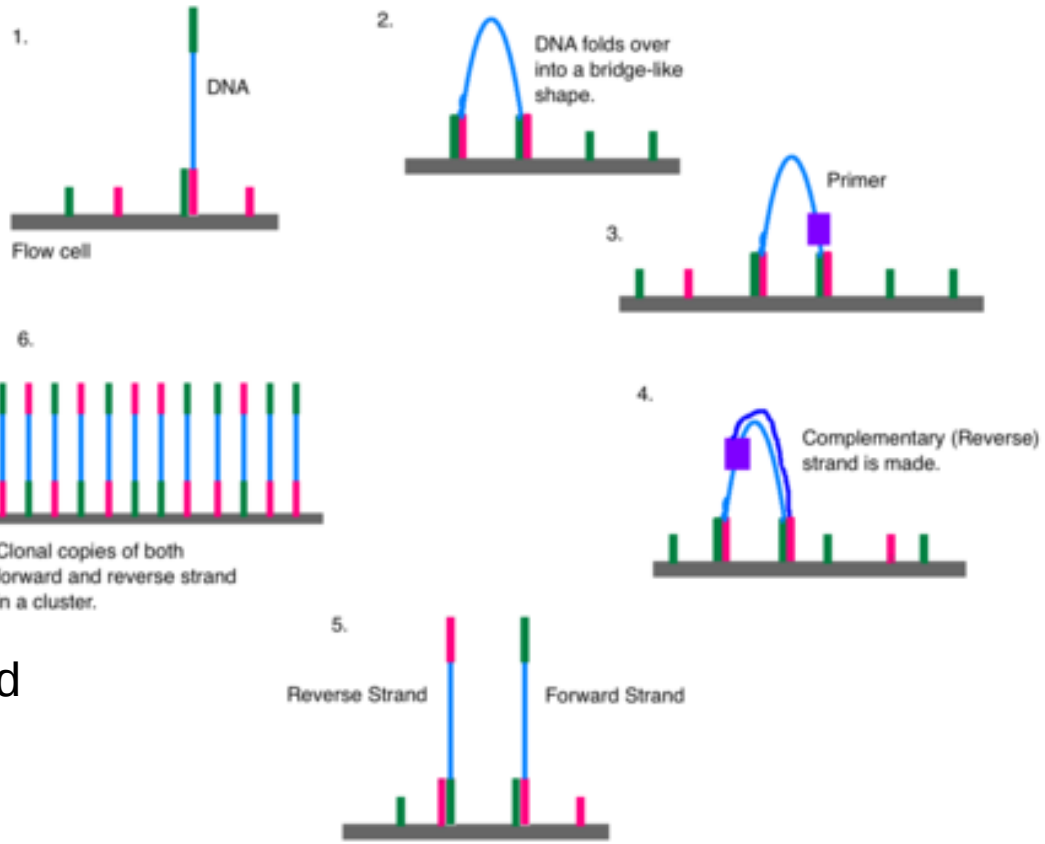


(and PCR amplification!)

library preparation

Illumina sequencing

1. Library preparation
2. ***Cluster generation via bridge amplification***
3. Sequencing by synthesis
4. Image capture
5. Convert to base calls



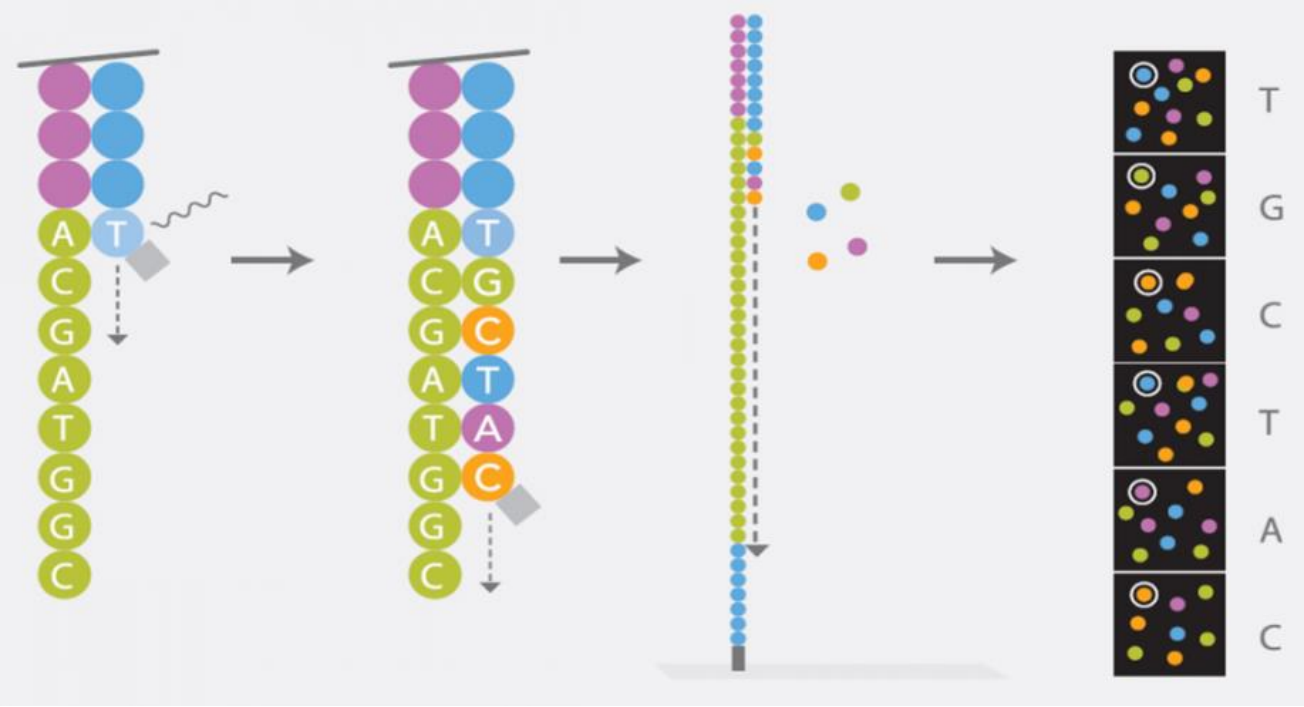
Short Illumina video
(https://tinyurl.com/hvnmwjb)

- Note
  - 2 PCR amplifications performed
    1. during ***library preparation***
    2. during ***cluster generation***
  - ***amplification always introduces bias!***

https://en.wikipedia.org/wiki/Illumina_dye_sequencing

Illumina sequencing

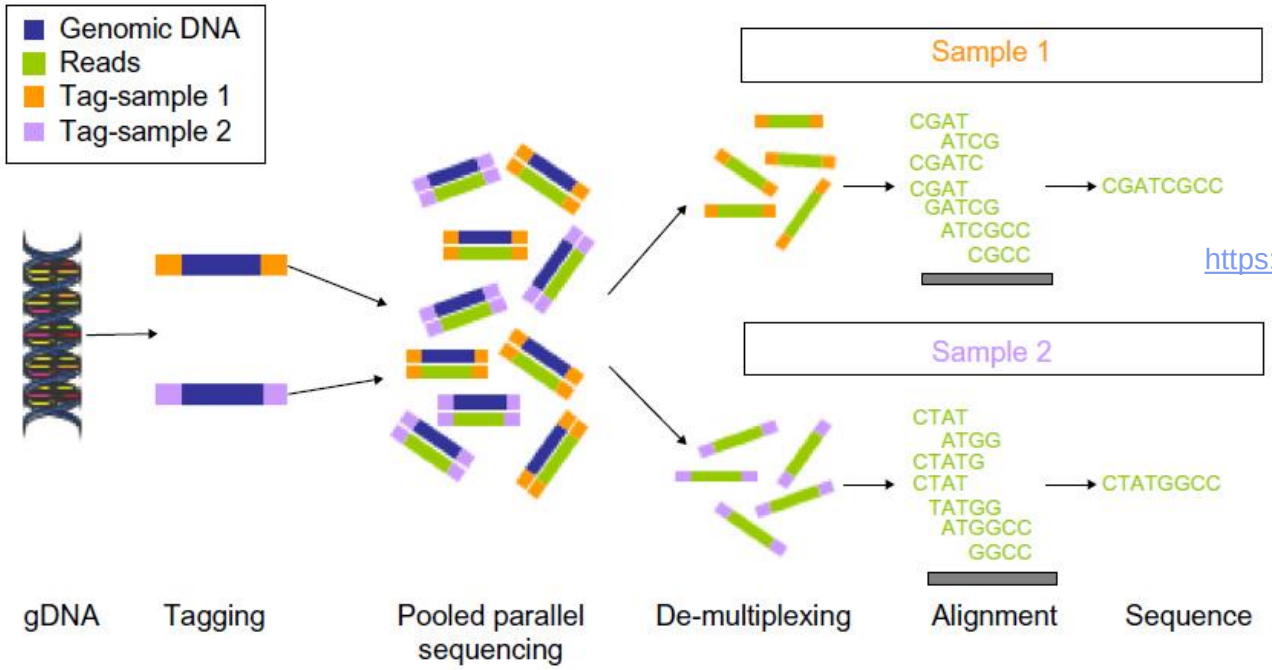
1. Library preparation
2. Cluster generation via bridge amplification
3. ***Sequencing by synthesis***
4. ***Image capture***
5. ***Convert to base calls***



http://www.cegat.de/

Multiplexing

- Illumina sequencers have one or more flowcell “lanes”, each of which can generate millions of reads
  - ~20**M** reads/lane for MiSeq, ~10**G** reads/lane for NovaSeq
- When less than a full flowcell lane is needed, multiple samples with different ***barcodes*** (a.k.a. ***indexes***) can be run on the same lane
  - 6-8 bp ***library barcode*** attached to DNA library fragments
  - data from sequencer must be ***demultiplexed*** to determine which reads belong to which library



https://doi.org/10.2147/BLCTT.S51503

Illumina sequencer models
(UT's sequencing core facility, GSAF)

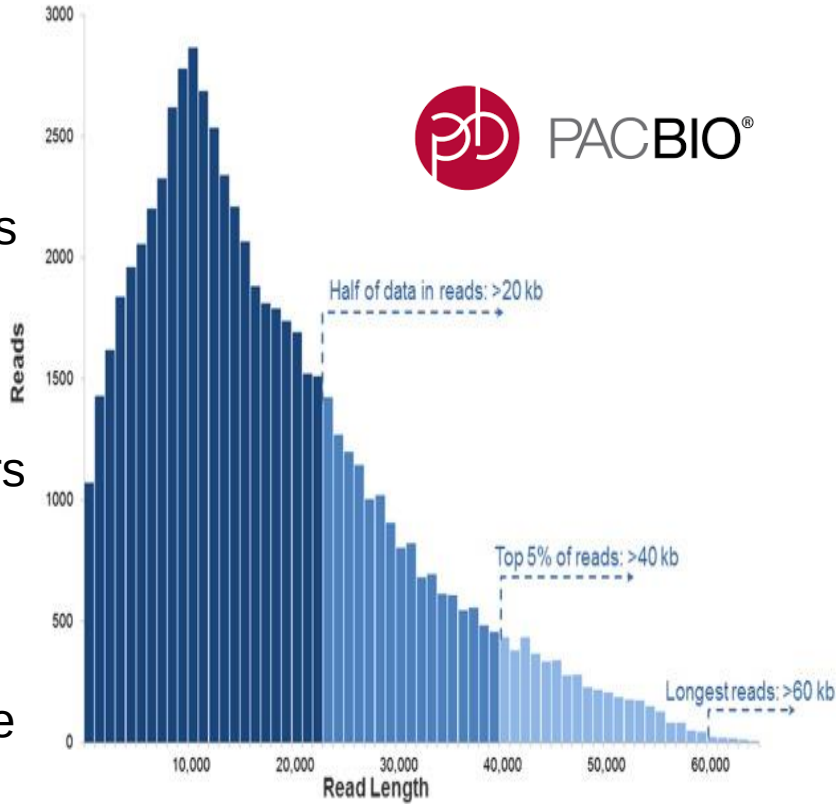
Model	Lanes	Typical reads per lane	Read lengths	Recommended applications
Nova Seq	2 (both get same DNA)	1 – 20 G	50, 100, 150, 250	WGS (Whole Genome Sequencing), WXS (Whole Exome Sequencing), RNA-seq, GBS (Genotyping by Sequencing) targeted sequencing
HiSeq 4000	8	240 M	50, 75, 150	
HiSeq 2500	8	200 M	36, 50, 75, 100, 125 (150, 250 rapid run)	
NextSeq	4 (all 4 get same DNA)	330 M	75, 150	
MiSeq	1	12 – 22 M (v2 vs v3 chemistry)	***v2:*** 25, 36, 150, 250 ***v3:*** 75, 300	Amplicons, metagenomics, WGS for tiny genomes, RNA-seq for small transcriptomes

Instrument cost: $125 K – $1+ M; Run cost: $1 K – $25 K

Long read sequencing

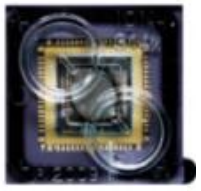
- Short read technology limitations
  - 30 – 300 base reads (150 typical)
  - PCR amplification bias
  - short reads are difficult to assemble
    - e.g. too short to span a long repeat region
  - difficult to detect large structural variations like inversions

- Newer "***single molecule***" sequencing
  - sequences ***single molecules***, not clusters
  - allows for ***much*** longer reads – multi-Kb!
    - no signal wash-out due to lack of synchronization among cluster molecules
  - ***but:*** weaker signal leads to high error rate
    - ~10+% *vs* <1% for Illumina
    - fewer reads are generated (~100 K)
  - one amplification usually still required (during library prep)



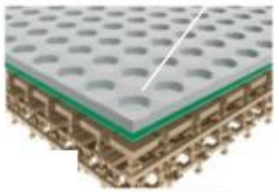
Long read sequencing

- Oxford Nanopore ION technology systems
  - https://nanoporetech.com/
  - DNA “spaghetti’s” through tiny protein pores
  - Addition of different bases produces different pH changes
    - measured as different changes in electrical conductivity
  - MinION is hand-held; starter kit costs ~$1,000 – including reagents!
    - inexpensive, but high error rates (~10%)



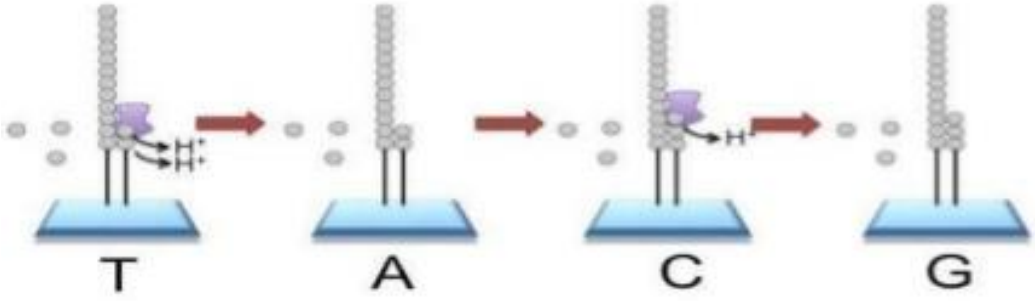
Chip

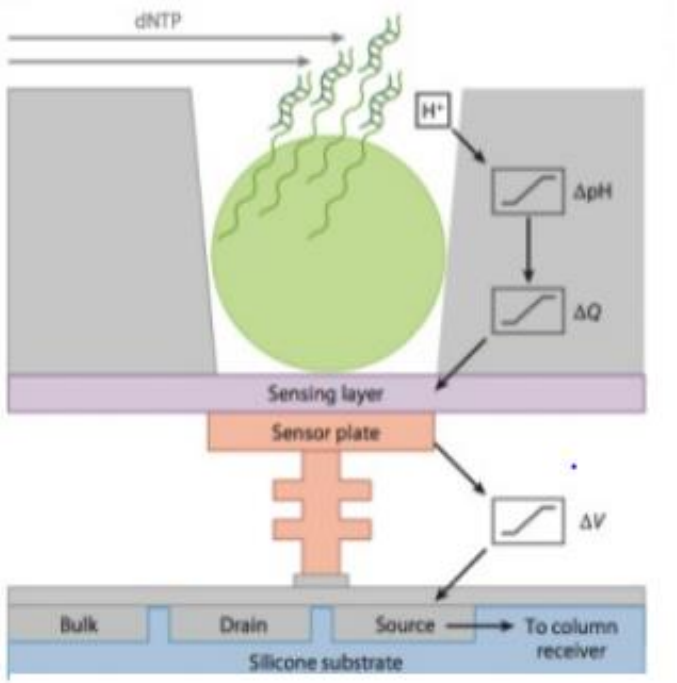
Semicor.ductor Packaging



Millions of Sensors

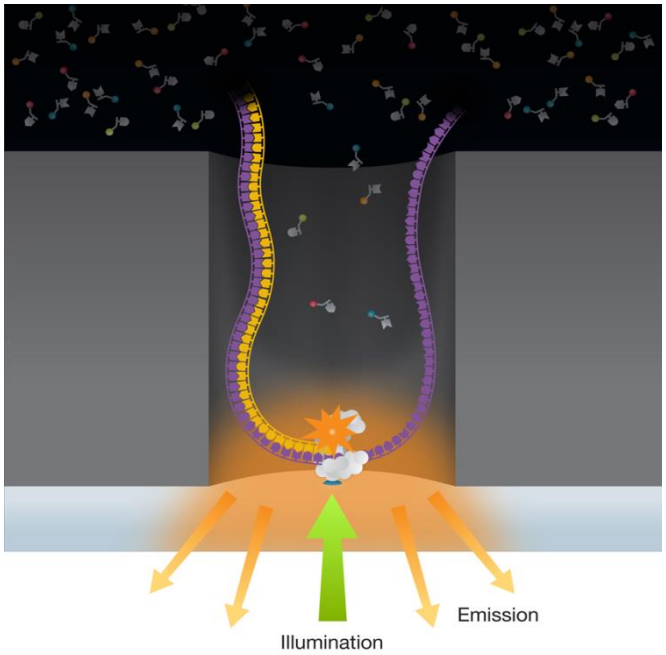
Semiconductor Design

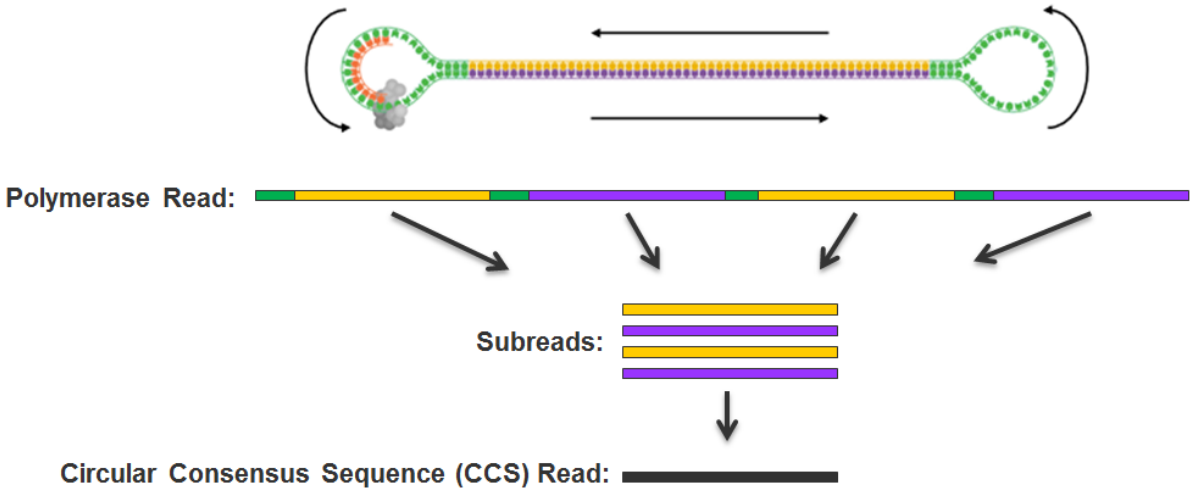




Long read sequencing

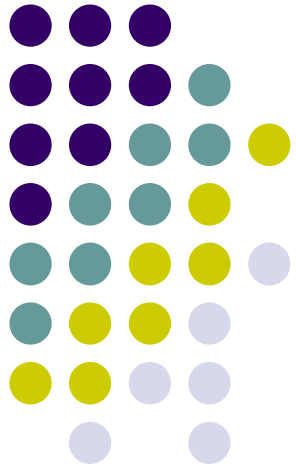
- PacBio SMRT system
  - http://www.pacb.com/smrt-science/smrt-sequencing/
  - Sequencing by synthesis in ***Zero-Mode Waveguide*** (ZMW) wells
  - DNA is circularized then repeatedly sequenced to achieve “consensus”
    - reduces error rate (~1-2%), but equipment *quite* expensive
  - Also have a PCR-free protocol (limited applications)

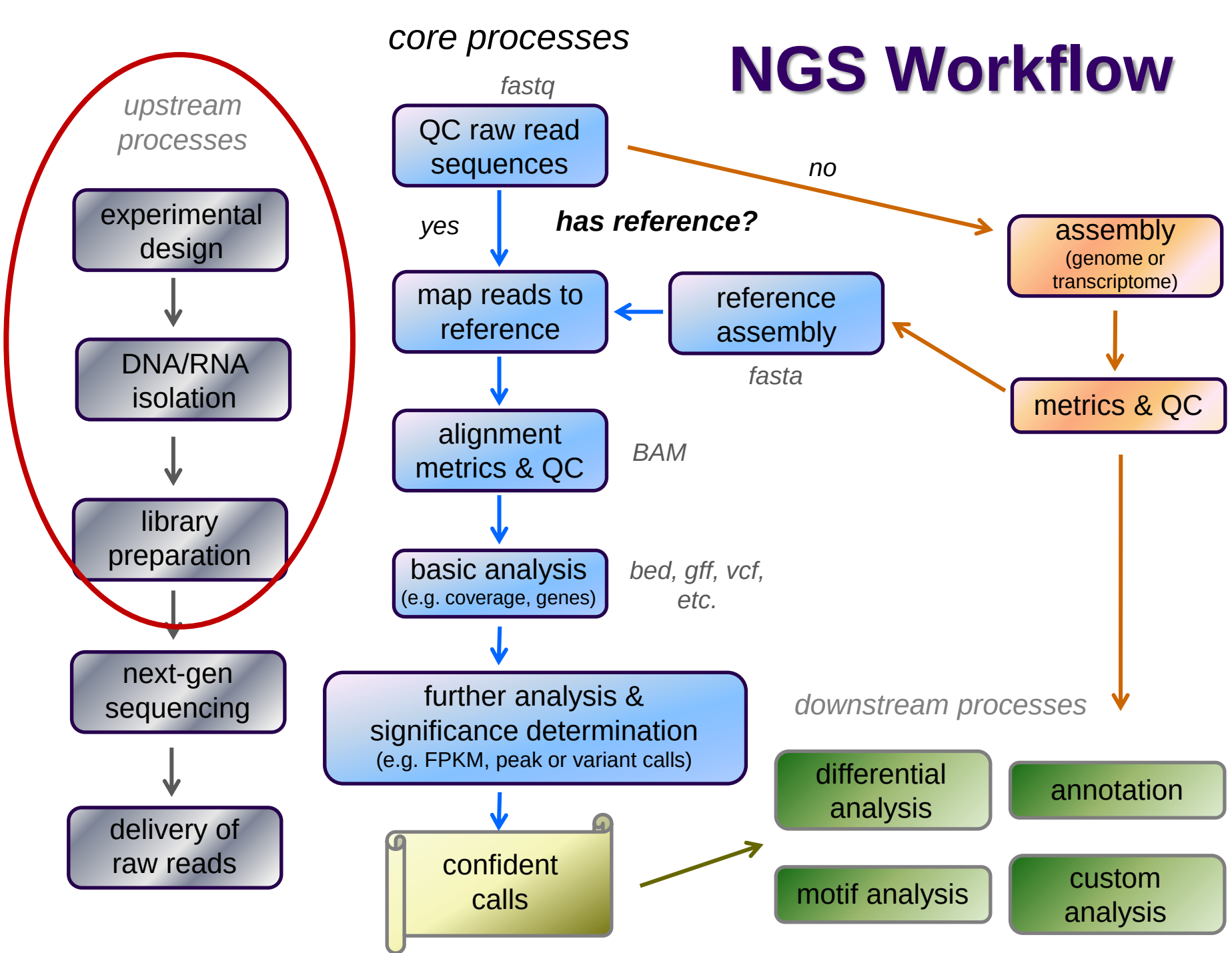




Part 2:
NGS Terminology

- Experiment types & library complexity
- Sequencing terminology
- Sequence duplication issues
- Molecular barcoding approaches




core processes
NGS Workflow
upstream processes
experimental design
DNA/RNA isolation
library preparation
next-gen sequencing
delivery of raw reads
fastq
QC raw read sequences
no
yes
has reference?
assembly
(genome or transcriptome)
map reads to reference
reference assembly
fasta
metrics & QC
alignment metrics & QC
BAM
basic analysis
(e.g. coverage, genes)
bed, gff, vcf, etc.
further analysis & significance determination
(e.g. FPKM, peak or variant calls)
downstream processes
confident calls
differential analysis
annotation
motif analysis
custom analysis

Library Complexity

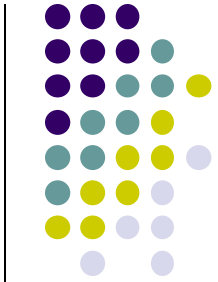
Library complexity (diversity) is a measure of the number of ***distinct molecular species*** in the library.

Many different molecules → ***high complexity***

Few different molecules → ***low complexity***

The number of different molecules in a library depends on ***enrichment*** performed during library construction.

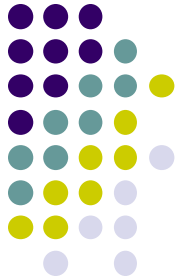
Popular Experiment Types



- ***Whole Genome sequencing (WGS)***
  - ***main application***: genome assembly
  - ***library***: all genomic DNA
  - ***complexity***: ***high*** (fragments must cover the entire genome)
- ***Exome sequencing (WXS)***
  - ***main application***: polymorphism/SNP detection; genotyping
  - ***library***: DNA from eukaryotic exon regions (uses special kits)
  - ***complexity***: ***high/med*** (only ~5% of eukaryotic genome is in exons)
- ***RNA-seq***
  - ***main application***: differential gene expression between 2 or more conditions
  - ***library***: extracted RNA converted to cDNA
  - ***complexity***: ***med/high*** (only a subset of genes are expressed in any given tissue)
- ***Amplicon panels (targeted sequencing)***
  - ***main applications***: genetic screening panels; metagenomics (e.g. 16S rRNA); mutagenesis
  - ***library***: DNA from a set of PCR-amplified regions using custom primers
  - ***complexity***: ***very low*** (only 1 to a few thousand different library molecules)

Type	Library construction	Applications	Complexity
Whole genome (WGS)	• extract genomic DNA & fragment	• Genome assembly • Variant detection, genotyping	high
Bisulfite sequencing	• bisulfite treatment converts C → U but not 5meC	• DNA Methylation profiling (CpG motifs)	high
RAD-seq, ddRAD	• restriction-enzyme digest DNA & fragment	• Variant detection (SNPs) • Population genetics, QTL mapping	high
Exome (WXS)	• capture DNA from exons only (manufacturer kits)	• Variant detection, genotyping	high-medium
ATAC-seq	• high-activity transposase cuts DNA & ligates adapters	• Profile nucleosome-free regions (“open chromatin”)	medium-high
RNA-seq, Tag-seq	• extract RNA & fragment • convert to cDNA (all fragments or just 3’ poly-A’d ends with Tag-seq)	• Differential gene or isoform expression • Transcriptome assembly (RNA-seq only)	medium, medium-low for Tag-seq
Transposon seq (Tn-seq)	• create library of transposon-mutated genomic DNA • amplify mutants via Tn-PCR	• Characterize genotype/phenotype relationships with high sensitivity	medium
ChIP-seq	• cross-link proteins to DNA • pull-down proteins of interest w/ specific antibody, reverse cross-links	• Genome-wide binding profiles of transcription factors, epigenetic marks & other proteins	medium (but variable)
GRO-seq	• isolate actively-transcribed RNA	• Characterize transcriptional dynamics	medium-low
RIP-seq	• like ChIP-seq, but with RNA	• Characterize protein-bound RNAs	low-medium
miRNA-seq	• isolate 15-25bp RNA band	• miRNA profiling	low
Amplicons	• amplify 1-1000+ genes/regions	• genotyping, metagenomics, mutagenesis	low

Library complexity is primarily a function of experiment type



higher complexity

Less enrichment for specific sequences

genomic

bisulfite-seq

exon capture

RNA-seq

ChIP-seq

amplicons

More enrichment for specific sequences

lower complexity

Higher diversity of library molecules
Lower sequence duplication expected
More sequencing depth required

… as well as…
- *genome size & sequencing depth*
- *library construction skill & luck!*

Lower diversity of library molecules
Higher sequence duplication expected
Less sequencing depth required

Read types

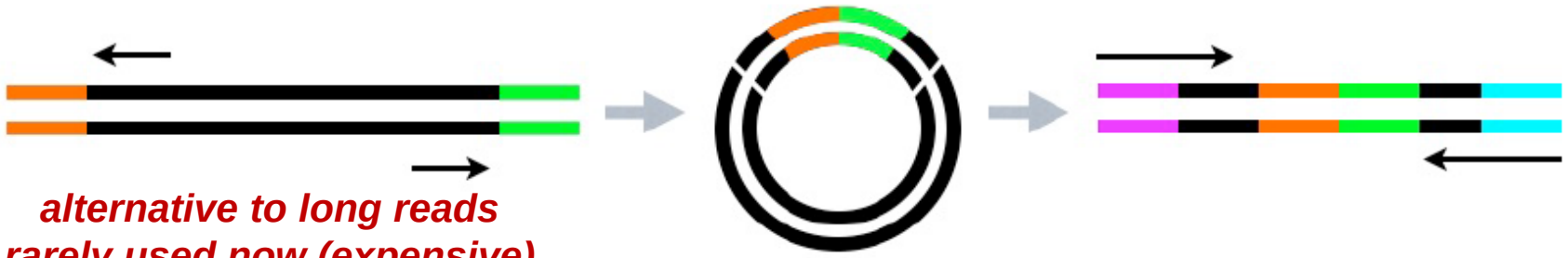
single-end

independent reads

paired-end

two inwardly oriented reads separated by ~200 nt

mate-paired



alternative to long reads
rarely used now (expensive)

two outwardly oriented reads separated by ~3000 nt

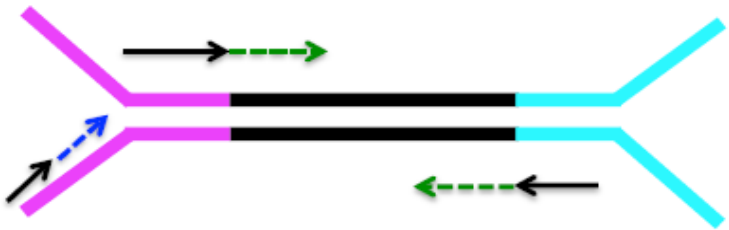
Read sequence terminology

Fragment library (input DNA sample)

Library prep

Sequencing library
Double-stranded or Y- **adaptors** added

DNA sequencing



Barcode (6–12 bases) – so many samples can be run in one lane. Data is **demultiplexed**.

Primers

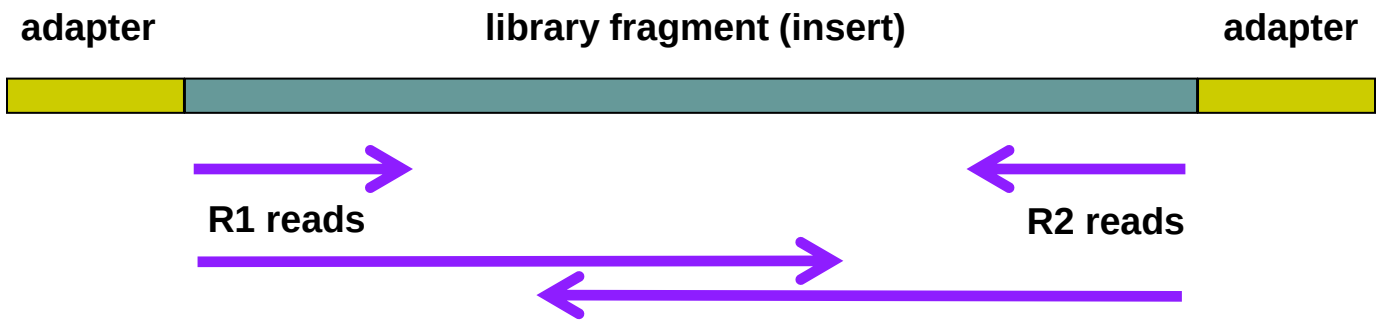
Reads (30 – 300 bases)

- Adapter areas include primers, barcode
  - sequencing facility will have more information

https://wikis.utexas.edu/pages/viewpage.action?pageId=28165137

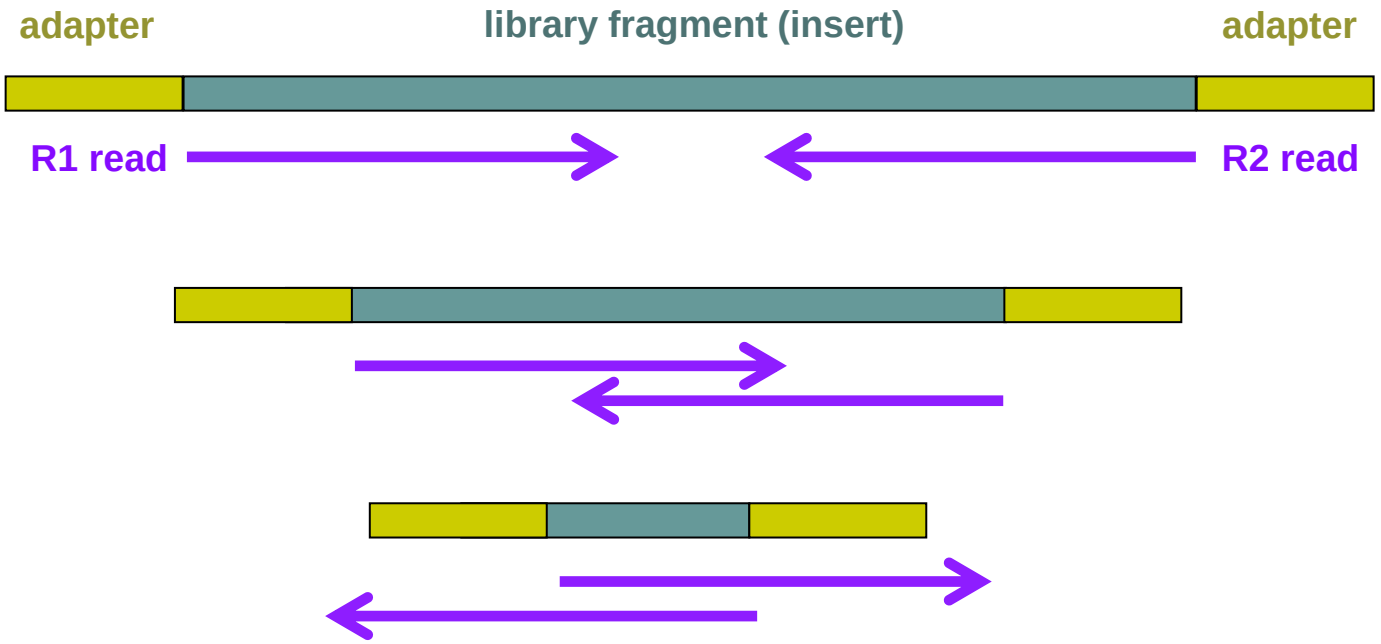
Reads and Fragments

- With paired-end (PE) sequencing, keep in mind the distinction between
  - the library ***fragment*** from your library that was sequenced
    - also called ***inserts***
  - the ***sequence reads*** (R1s & R2s) you receive
    - also called ***tags***
  - an R1 and its associated R2 form a ***read pair***
    - a readout of part (or all) of the fragment molecule
- There is often confusion of terminology in this area!
  - Be sure to request depth in ***read pairs*** for paired-end sequencing



Library fragment distribution

- Fixed size in your sequencing library:
  - the adapter region (including all barcodes)
  - the read length (e.g. 50, 100, 150)
- But the insert fragments are of variable length
  - due to random shearing during sonication
  - bioanalyzer gives an idea of your library's fragment distribution



Single end *vs* Paired end

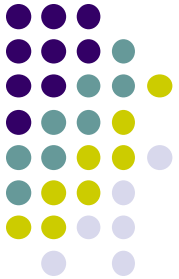
- ***single end*** (SE) reads are less expensive
  - ***but*** SE reads provide less information
- ***paired end*** (PE) reads can be mapped more reliably
  - especially against lower complexity genomic regions
    - an unmapped read can be “rescued” if its mate maps well
  - they provide more bases around a locus
    - e.g. for analysis of polymorphisms

  - actual fragment sizes can be easily determined
    - from the alignment records for each dual-mapping “proper pair”
  - also help distinguish the true complexity of a library
    - by clarifying which ***fragments*** are duplicates (vs ***read*** duplicates)
  - ***but*** PE reads are more expensive – and more data
    - more storage space and processing time required
- General guidelines
  - use PE for high location accuracy and/or base-level sensitivity
  - use SE for lower-complexity, higher duplication experiments

Sequencing depth

- How much sequencing depth is needed?
  - No single answer! Consult your sequencing facility.
- Depends on:
  - genome size
    - prokaryotes – up to a few Megabases (*E. coli*: 5 Mbase)
    - lower eukaryotes – 10+ Megabases (yeast: 12 Mbase; worm 100 Mbase)
    - higher eukaryotes – Gigabases (chicken: 1 Gbase; human: 3 Gbase)
  - theoretical library complexity / library fragment enrichment
    - genomic re-sequencing ***vs*** amplicon sequencing
    - total RNA-seq ***vs*** 3' Tag-seq
    - ChIP-seq ***vs*** RIP-seq
  - desired sensitivity
    - e.g. looking for rare mutations

Sequencing depth required is a function of experiment type & genome size



higher complexity

Less enrichment for specific sequences

genomic

bisulfite-seq

exon capture

RNA-seq

ChIP-seq

amplicons

More enrichment for specific sequences

lower complexity

Higher diversity of library molecules
Lower sequence duplication expected
More sequencing depth required

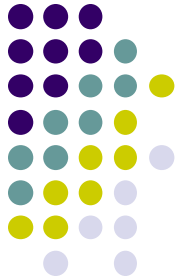
… and more depth is required for larger genomes

Lower diversity of library molecules
Higher sequence duplication expected
Less sequencing depth required

Sequence Duplication

- The set of sequences you receive can contain ***exact duplicates***
- Duplication can arise from:
  1. sequencing of species enriched in your library (***biological – good!***)
     - each read comes from a different DNA molecule (cluster)
  2. sequencing of artifacts (***technical – bad!***)
     - differentially amplified PCR species (PCR duplicates)
       - recall that 2 PCR amplifications are performed with Illumina sequencing
     - optical duplicates, when two Illumina flowcell clusters overlap
  - ***cannot tell which using "standard" sequencing methods!***
- Standard best practice is to "mark duplicates" during initial processing
  - then decide what to do with them later…
    - e.g. retain (use all), remove (use only non-duplicates), dose (use some)
- Different experiment types have different ***expected*** duplication
  - whole genome/exome → high complexity & low duplication
  - amplicon sequencing → low complexity & high duplication

Expected sequence duplication is primarily a function of experiment type



higher complexity

Less enrichment for specific sequences

genomic

bisulfite-seq

exon capture

RNA-seq

ChIP-seq

amplicons

More enrichment for specific sequences

lower complexity

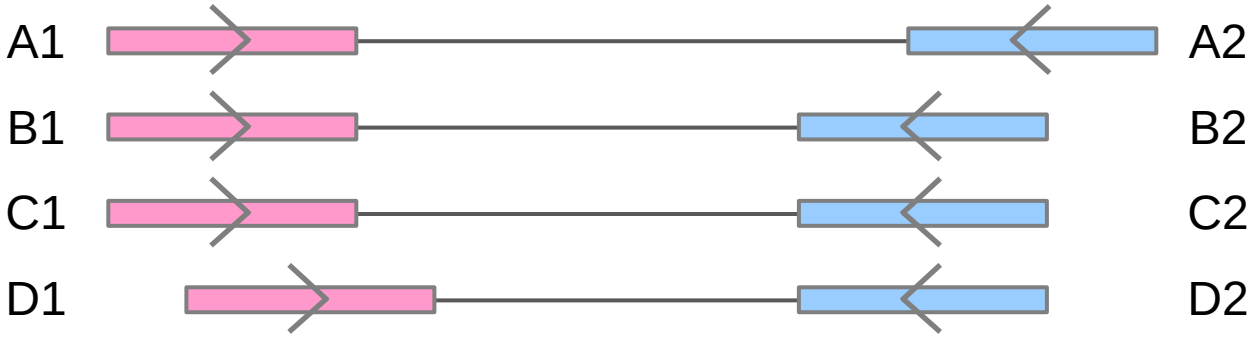
Higher diversity of library molecules
Lower sequence duplication expected
More sequencing depth required

… as well as…
- *genome size & sequencing depth*
- *library construction skill & luck!*

Lower diversity of library molecules
Higher sequence duplication expected
Less sequencing depth required

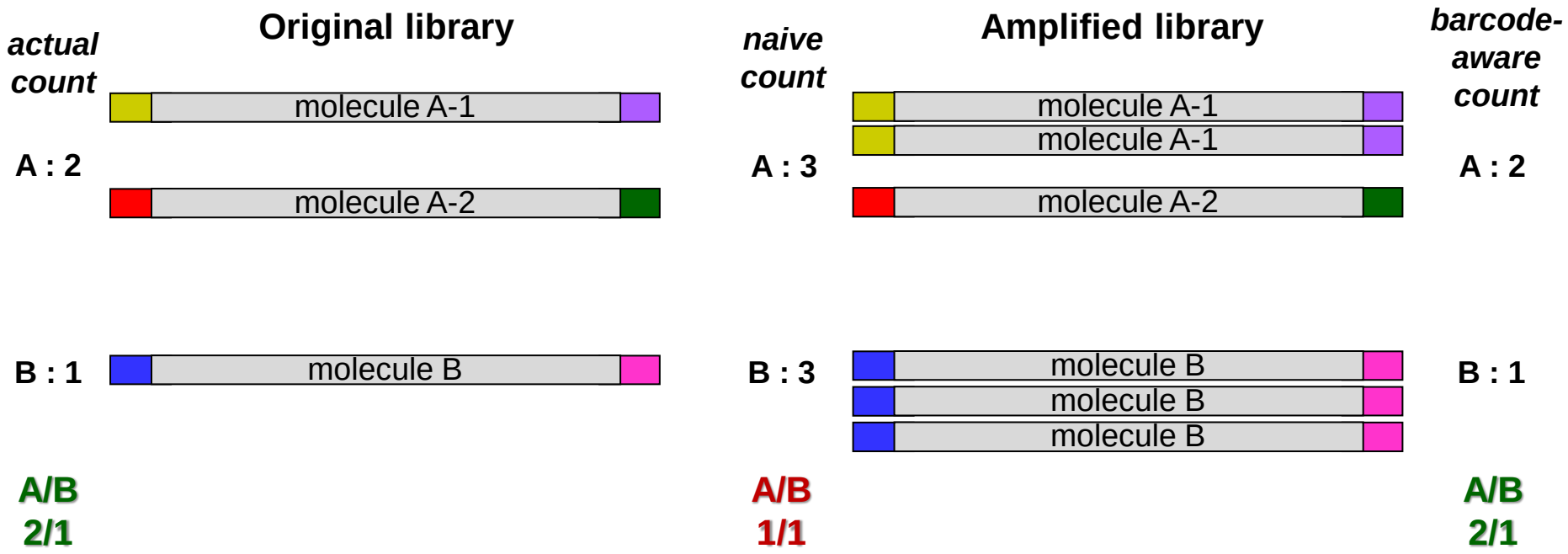
Read vs Fragment duplication

- Consider the 4 fragments below
  - 4 R1 reads (pink), 4 R2 reads (blue)
- Duplication when only 1 end considered
  - A1, B1, C1 have identical sequences, D1 different
    - 2 unique + 2 duplicates = 50% duplication rate
  - B2, C2, D2 have identical sequences, A2 different
    - 2 unique + 2 duplicates = 50% duplication rate
- Duplication when both ends considered
  - fragments B and C are duplicates (same external sequences)
    - 3 unique + 1 duplicate = 25% duplication rate

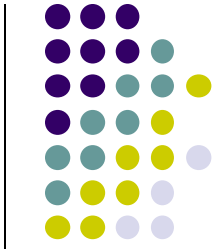


Molecular Barcoding

- Resolves ambiguity between biological and technical (PCR amplification) duplicates
  - adds secondary ***internal barcodes*** to ***pre-PCR*** molecules
    - a.k.a. ***UMI****s (**U**nique **M**olecular **I**ndexes)*
  - combination of barcodes + insert sequence provides accurate quantification
  - but requires specialized library prep & computational post-processing
    - e.g. 3' Tag-seq tag de-duplication; scRNA-seq UMI de-duplication



Single *Cell* sequencing



- Standard sequencing library starts with millions of cells
  - will be in different states unless synchronized
  - a heterogeneous “ensemble” with (possibly) high cell-to-cell variability
- ***Single cell sequencing*** technologies aim to capture this variability
  - examples:
    - cells in different layers/regions of somatic tissue (identify novel cell subtypes)
    - cells in different areas of a tumor (identify “founder” mutations)
  - essentially a very sophisticated library preparation technique
- Typical protocol (RNA-seq)
  1. isolate a few thousand cells (varying methods, e.g. FACs sorting, cryostat sectioning)
  2. the single-cell platform partitions each cell into an emulsion droplet
     - e.g. 10x Genomics (https://www.10xgenomics.com/solutions/single-cell/)
  3. a different barcode (UMI) is added to the RNA in each cell
  4. resulting library submitted for standard Illumina short-read sequencing
  5. custom downstream analysis links results to their cell (barcode) of origin

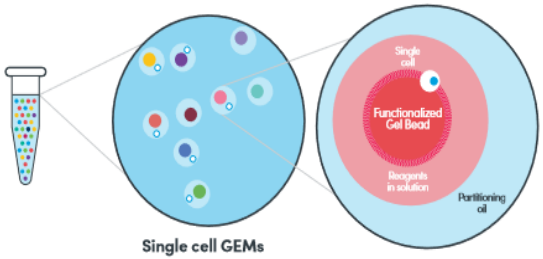
10x Genomics Chromium

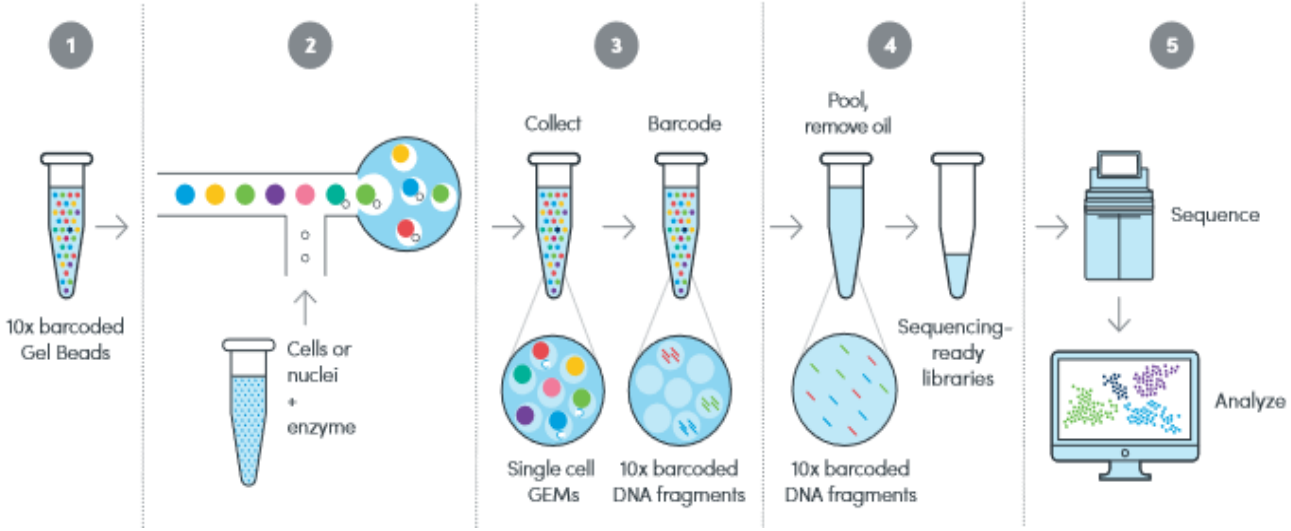
Next GEM technology

1. Every Chromium solution starts with a high-diversity pool of Gel Beads, each coated with a unique oligonucleotide barcode sequence, and functionalized sequences to capture molecules of interest.
2. Within the Chromium instrument, barcoded Gel Beads are mixed with cells or nuclei, enzymes, and partitioning oil to form tens of thousands of single cell emulsion droplets called "GEMs" (Gel Bead-in-emulsion).
3. Each GEM acts as an individual reaction droplet in which the Gel Beads are dissolved and molecules of interest from each cell are captured and barcoded.
4. After barcoding, all fragments from the same cell or nucleus share a common 10x Barcode. Barcoded fragments for hundreds to tens of thousands of cells are pooled for downstream reactions to create short-read sequencer compatible libraries.
5. After sequencing, turnkey bioinformatics tools use the identifying barcodes to map sequencing reads back to their single cell or nucleus of origin.

A GEM is a "Gel Bead-in-emulsion" droplet that encapsulates each micro-reaction within the Chromium instrument.

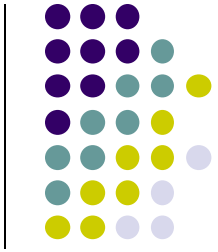
Here, we show a GEM with a single cell, reagents, and barcoded Gel Bead all partitioned within a single droplet.





https://pages.10xgenomics.com/rs/446-PBO-704/images/10x_BR025_Chromium-Brochure_Letter_Digital.pdf

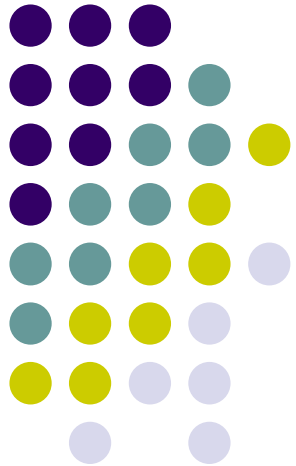
Some barcode (index) types

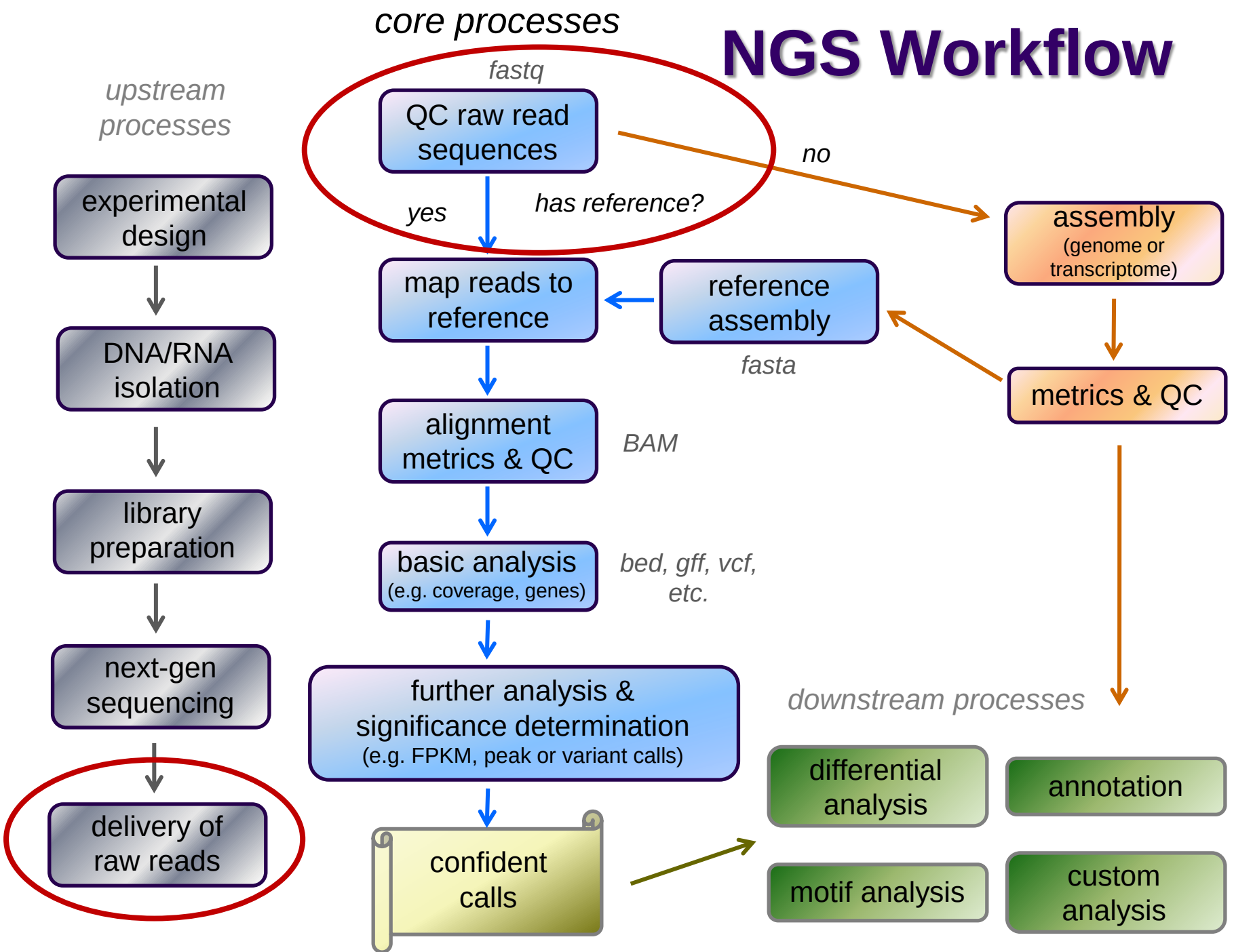


- ***Library barcode***
  - multiple barcoded samples can be pooled on one sequencer lane
  - Index is the same for all fragments in a library
    - ~100 available (part of standard library prep kits)
- ***Molecular barcodes (Unique Molecular Index, UMI)***
  - added to achieve accurate fragment quantification (e.g. 3' Tag-seq)
    - addresses ambiguity between biological and technical sequence duplication
  - different, small barcodes (or pairs) attached to library fragments ***before PCR amplification***
  - available diversity depends on barcode size and number, e.g.:
    - 4 well-separated bases → ~80; 2 x 4 well-separated bases → ~700; 2 x 8 well-separated bases → ~500,000
      - finding well-separated, sequencing-compatible barcodes is not trivial!
- ***Single cell molecular barcode***
  - UMI attached to all cDNA molecules in *each single cell*
    - number of barcodes needed depends on # of single cells desired

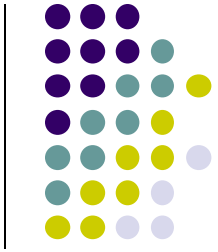
Part 3:
The FASTQ format, Data QC & preparation

- FASTA and FASTQ formats
- QC of raw sequences with **FastQC** tool
- Dealing with adapters




core processes
NGS Workflow
upstream processes
fastq
QC raw read sequences
no
yes
has reference?
experimental design
assembly (genome or transcriptome)
map reads to reference
reference assembly
fasta
DNA/RNA isolation
metrics & QC
alignment metrics & QC
BAM
library preparation
basic analysis (e.g. coverage, genes)
bed, gff, vcf, etc.
next-gen sequencing
further analysis & significance determination (e.g. FPKM, peak or variant calls)
downstream processes
differential analysis
annotation
delivery of raw reads
confident calls
motif analysis
custom analysis

FASTQ files



- Nearly all sequencing data delivered as ***FASTQ*** files
  - **FASTQ** = **FASTA** sequences + **Q**uality scores
  - file names have **.fastq** or **.fq** extensions
  - usually compressed to save space
    - (**gzip**'d, with **.gz** file extension)
    - best practice: leave them that way!
      - 3x to 6x space saving
      - most tools handle **gzip**'d FASTQ
- Paired-end sequencing data comes in 2 FASTQs
  - one each for R1 and R2 reads, same number of rows

    Sample_*MyTubeID*_L008_R1.fastq.gz

    Sample_*MyTubeID*_L008_R2.fastq.gz
  - ***order of reads is identical***
    - aligners rely on this "***name ordering***" for paired-end alignment

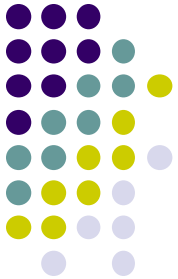
FASTQ format

- Text format for storing sequence and quality data
  - http://en.wikipedia.org/wiki/FASTQ_format
- 4 lines per sequence:
  1. ***@read name*** *(plus extra information after a space)*
     - ***R1 and R2 reads have the same read name***
  2. ***called base sequence*** **(ACGTN)**

     always 5' to 3'; *usually* excludes 5' adapter
  3. ***+optional other information***
  4. ***base quality scores encoded as text characters***
- FASTQ representation of a single, 50 base R2 sequence

```
@HWI-ST1097:97:D0WW0ACXX:4:1101:2007:2085 2:N:0:ACTTGA
ATTCTCCAAGATTTGGCAAATGATGAGTACAATTATATGCCCCAATTTACA
+
??@@?DD;?;FF?HHBB+:ABECGHDHDCF4?FGIGACFDFH;FHEIIIB9?
```

FASTQ read names



- Illumina FASTQ read names encode information about the source cluster
  - unique identifier ("fragment name") begins with @, then:
    - sequencing machine name + flowcell identifier
    - lane number
    - flowcell coordinates
  - a space separates the name from extra read information:
    - end number (**1** for R1, **2** for R2)
    - two quality fields (N = ***Not*** QC failed)
    - barcode sequence
  - R1, R2 reads ***have the same fragment name***
    - this is how the reads are linked to model the original fragment molecule

```
@HWI-ST1097:97:D0WW0ACXX:4:1101:2007:2085  1:N:0:ACTTGA
@HWI-ST1097:97:D0WW0ACXX:4:1101:2007:2085  2:N:0:ACTTGA
```

FASTQ quality scores

- Base qualities expressed as ***Phred*** scores
  - ***Phred*** scores are log scaled ***higher = better***
    - *versus* probability [0,1] ***P-value***, where ***lower = better***
  - ***Quality***: 20 = $1.0e^{-2}$ = 1/100 errors; 30 = $1.0e^{-3}$ = 1/1000 errors

$$\textbf{Probability of Error} = 10^{-Q/10}$$

- Integer Phred score converted to Ascii character (add 33)

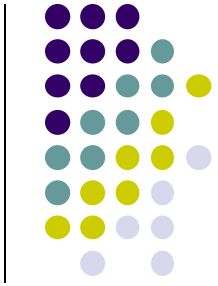
https://www.asciitable.com/

```
Quality character   !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJ
                    |          |          |          |          |
ASCII Value         33         43         53         63         73
Base Quality (Q)    0          10         20         30         40
```

```
?@@?DD;?;FF?HHBB+:ABECGHDHDCF4?FGIGACFDFH;FHEIIIB9?
```

In older Illumina/Solexa FASTQ files, ASCII offsets may differ
modern ***Sanger*** format shown above
see http://en.wikipedia.org/wiki/FASTQ_format for others

Multiple lanes



- One submitted sample may be delivered as multiple FASTQ files

  Lane1: **Sample_*MyTubeID*_L001_R1.fastq.gz, Sample_*MyTubeID*_L001_R2.fastq.gz**

  Lane2: **Sample_*MyTubeID*_L002_R1.fastq.gz, Sample_*MyTubeID*_L002_R2.fastq.gz**

  - NovaSeq always runs samples on both lanes; NextSeq on all 4 lanes
  - sometimes the sequencing facility splits your sample across lanes

- Your sample may be re-run to "top off" requested read depth
  - be careful with the file names!
    - if run in the same lane, the FASTQ file names will be ***identical***

  1st run: **Sample_*MyTubeID*_L003_R1.fastq.gz**

  2nd run : **Sample_*MyTubeID*_L003_R1.fastq.gz**

- Best practice
  - keep original data in separate directories by date & project
  - process data from multiple lanes separately for as long as possible
    - e.g. through FASTQ quality assurance
    - allows detection of lane-specific artifacts or anomalies

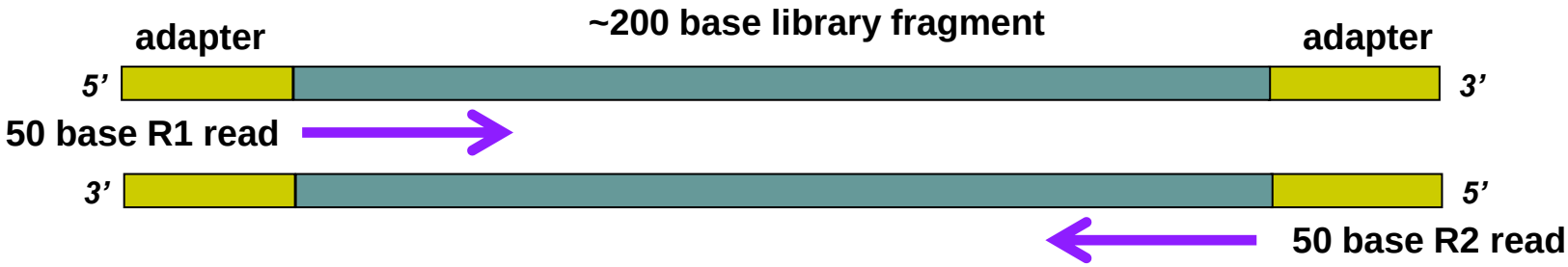
Raw sequence quality control

- Critical step! Garbage in → Garbage out
  - general sequence quality metrics
    - base quality distributions
    - sequence duplication rate
  - trim 3' adapter sequences?
    - important for RNA-seq
  - trim 3' bases with poor quality?
    - important for *de novo* assembly
  - other contaminants?
    - biological – rRNA in RNA-seq
    - technical – samples sequenced w/other barcodes

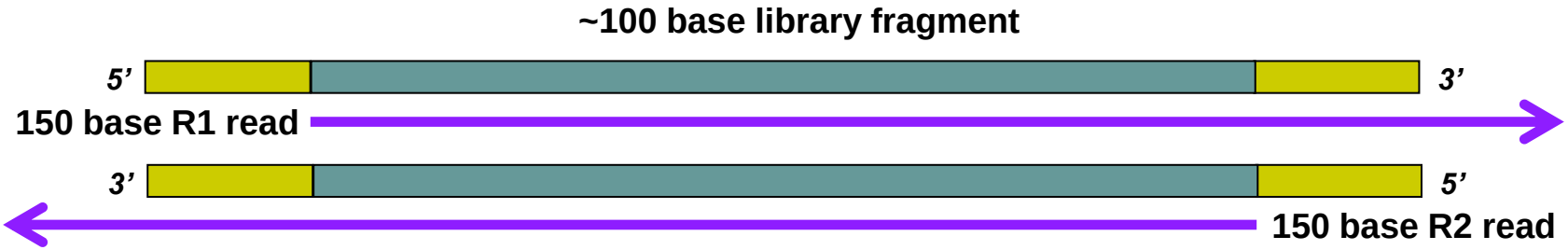
- Know your data
  - sequencing center pre-processing
    - 5' adapter removed? QC-failed reads filtered?
  - PE reads? relative orientations? molecular barcodes present?
  - technology specific issues?
    - e.g. bisulfite sequencing should produce C→T transitions

3’ Adapter contamination

A. reads short compared to fragment size (no contamination)

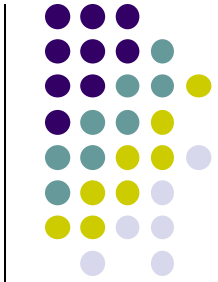


B. Reads long compared to library fragment (3’ adapter contamination)



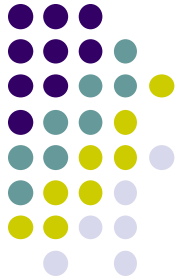
The presence of the 3’ adapter sequence in the read can cause problems during alignment, because it does not match the genome.

FastQC



- Quality Assurance tool for FASTQ sequences
- Can run as interactive tool or command line
- Input:
  - FASTQ file(s)
  - run on both R1, R2 files
- Output:
  - directory with html & text reports
    - **fastqc_report.html**
    - **fastqc_data.txt**

Most useful FastQC reports



1. *Per-base sequence quality Report*

   based on *all* sequences

   - Should I trim low quality bases?

2. *Sequence duplication levels Report*

   estimate based on *1st 100,000* sequences, trimmed to 50bp

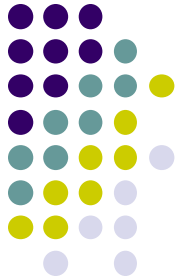
   - How complex is my library?

3. *Overrepresented sequences Report*

   based on *1st 100,000* sequences, trimmed to 75bp

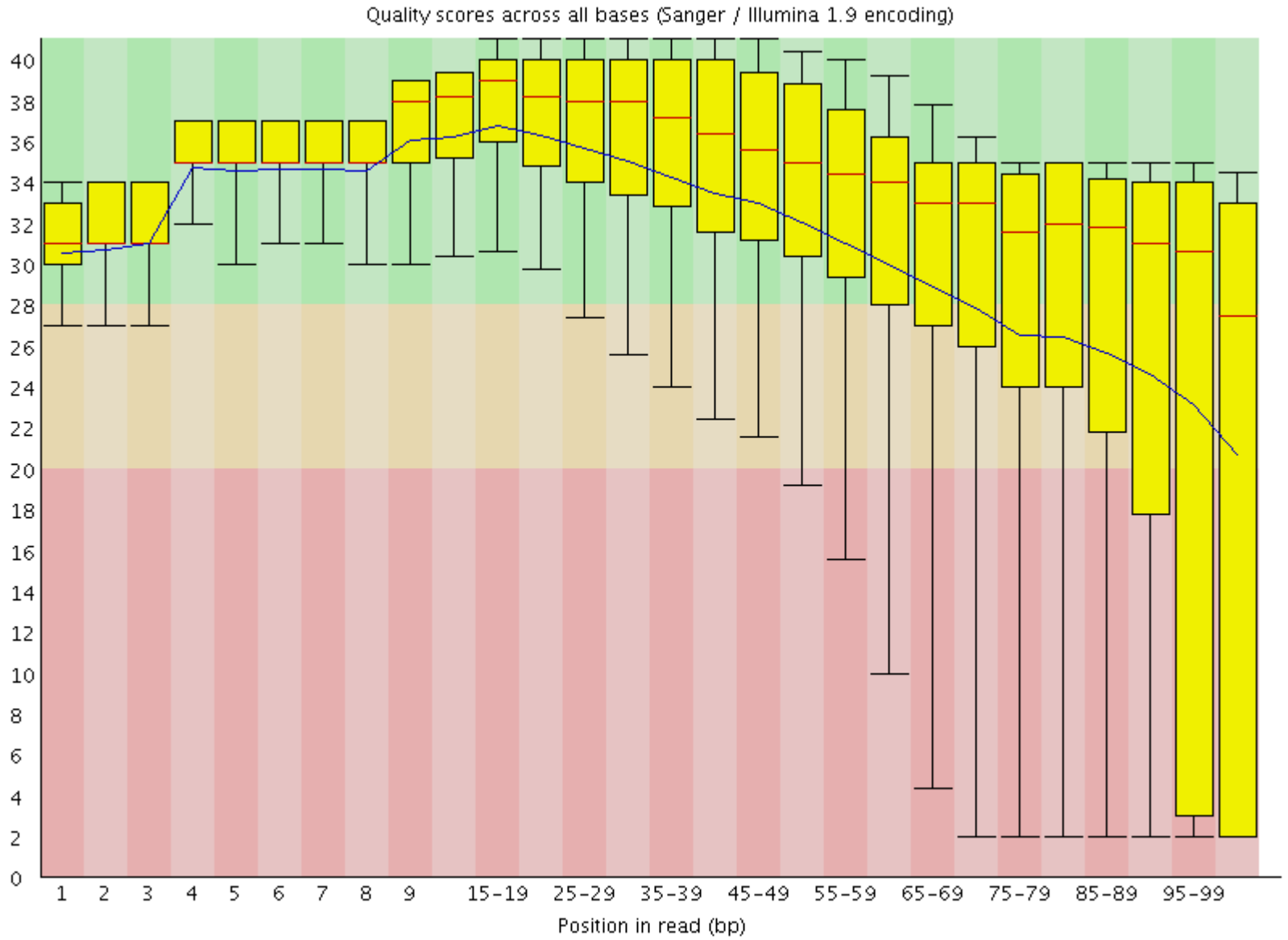
   - Do I need to remove adapter sequences?

FastQC resources



- FastQC website:
  http://www.bioinformatics.babraham.ac.uk
- FastQC report documentation:
  http://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/3%20Analysis%20Modules/
- Good Illumina dataset:
  http://www.bioinformatics.babraham.ac.uk/projects/fastqc/good_sequence_short_fastqc/fastqc_report.html
- Bad Illumina dataset:
  http://www.bioinformatics.babraham.ac.uk/projects/fastqc/bad_sequence_fastqc/fastqc_report.html
- Real Yeast ChIP-seq dataset:
  http://web.corral.tacc.utexas.edu/BioITeam/yeast_stuff/Sample_Yeast_L005_R1.cat_fastqc/fastqc_report.html

FastQC Per-base sequence quality report



FastQC Sequence duplication report

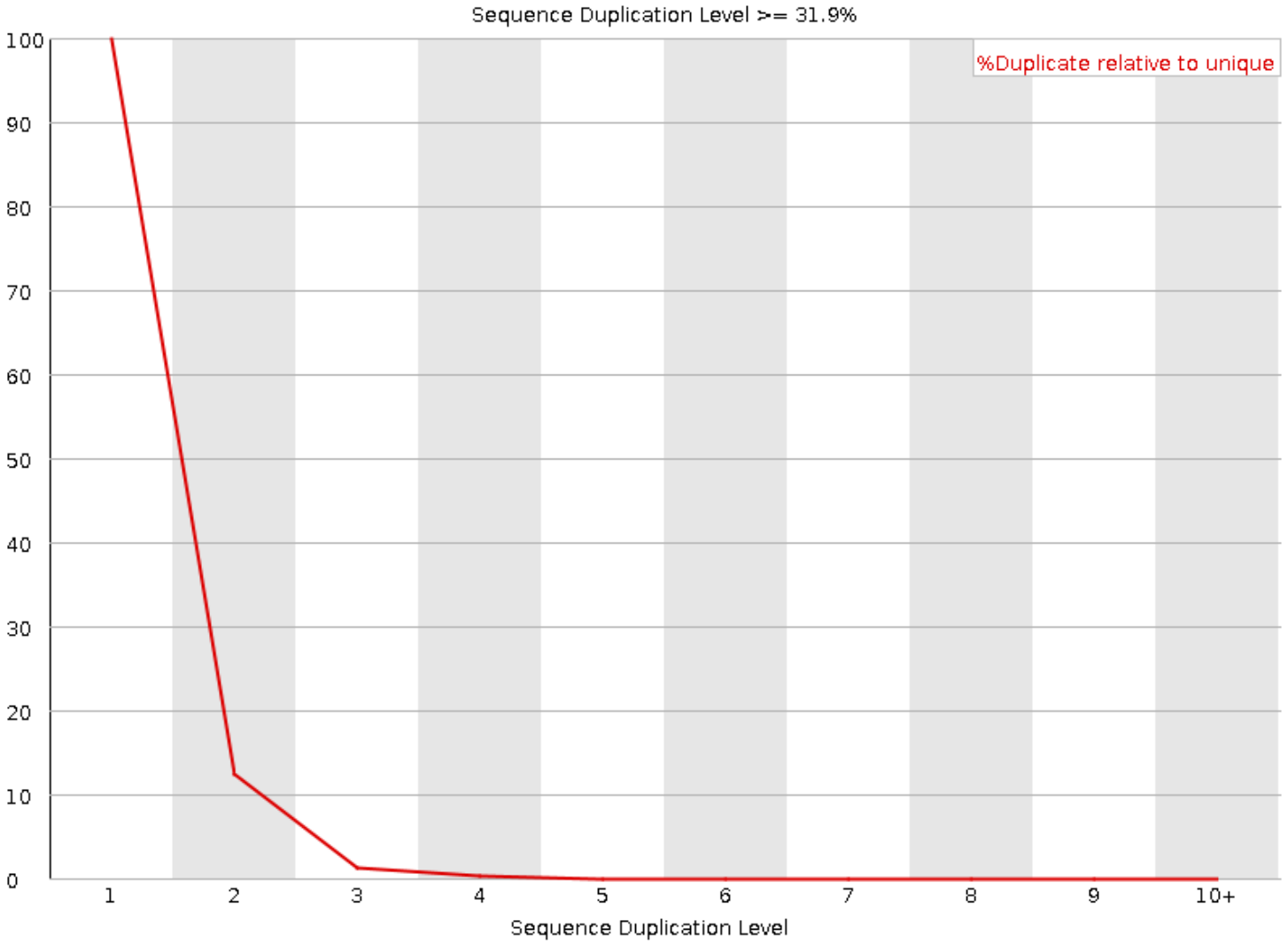
Yeast ChIP-seq

For every 100 unique sequences there are:

~12 sequences w/2 copies

~1-2 with 3 copies

Ok – Some duplication expected due to IP enrichment



Sequence duplication report

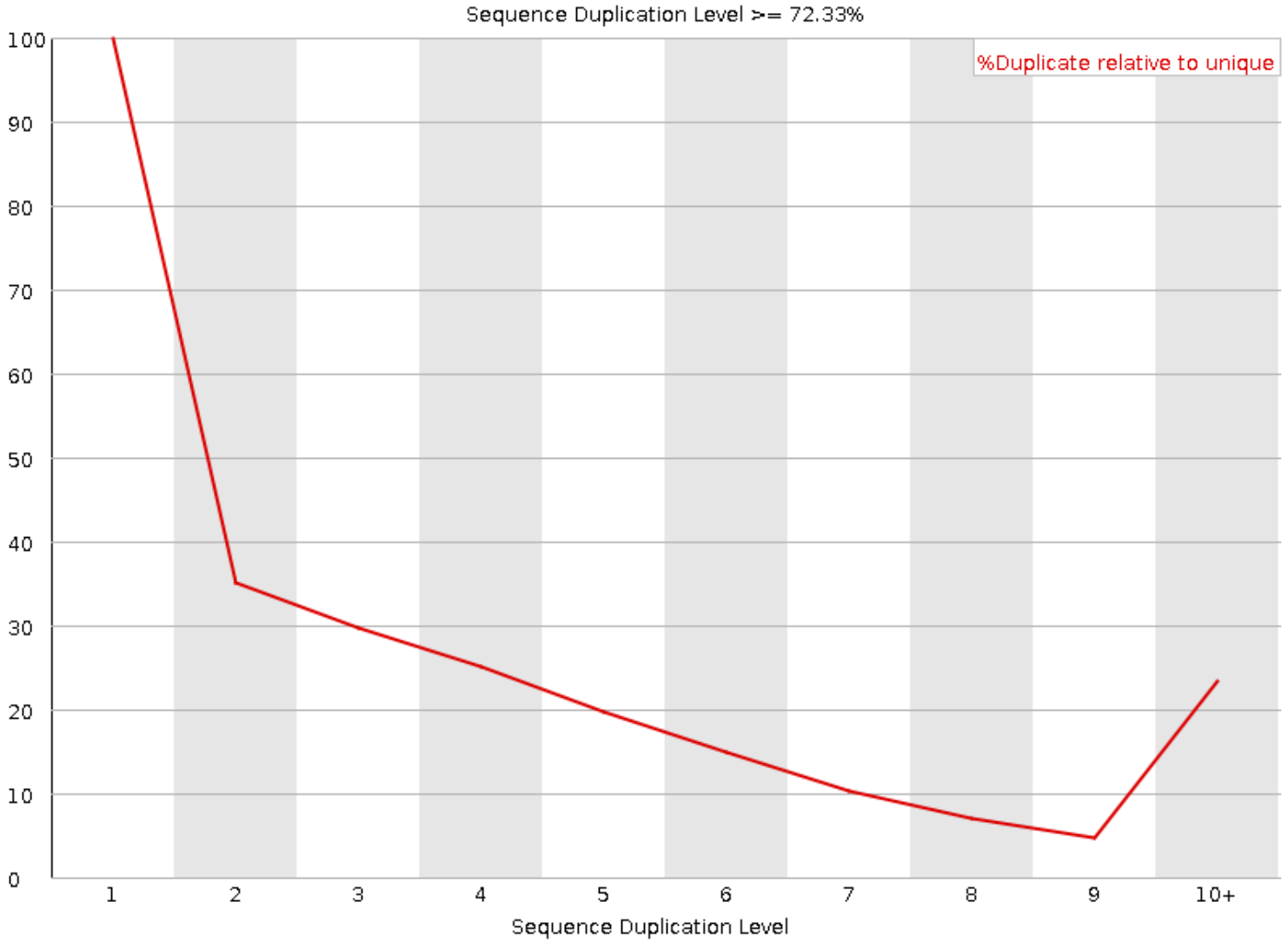
Yeast ChIP-exo

For every 100 unique sequences there are:

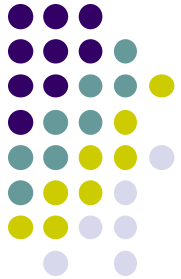
~35 sequences w/2 copies

~22 with 10+ copies

Success! Protocol expected to have high duplication



Expected sequence duplication is primarily a function of experiment type



higher complexity

Less enrichment for specific sequences

genomic

bisulfite-seq

exon capture

RNA-seq

ChIP-seq

ChIP-exo

amplicons

More enrichment for specific sequences

lower complexity

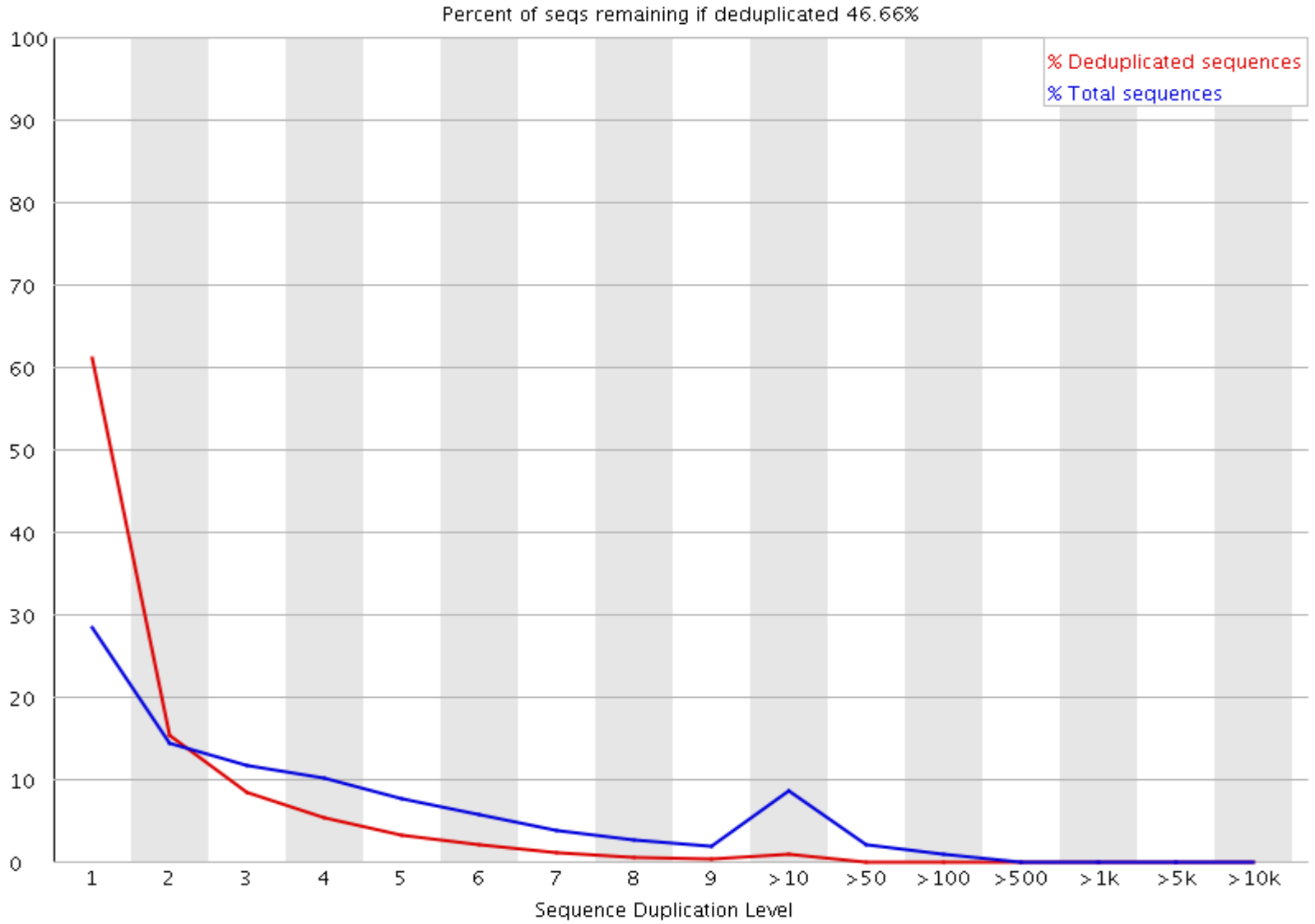
Higher diversity of library molecules
Lower sequence duplication expected
More sequencing depth required

… as well as…
- *genome size & sequencing depth*
- *library construction skill & luck!*

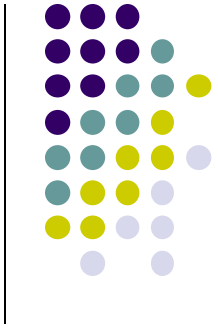
Lower diversity of library molecules
Higher sequence duplication expected
Less sequencing depth required

Newer FastQC versions have a slightly different Sequence Duplication report

- Red ***"deduplicated"*** line as previously described
- Blue ***"total"*** line is percentage histogram



FastQC Overrepresented sequences report



- **FastQC** knows Illumina adapter sequences
- Here ~9-10% of sequences contain adapters
  - calls for adapter removal or trimming

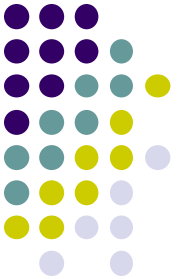
Sequence	Count	Percentage	Possible Source
AGATCGGAAGAGCACACGTCTGAACTCCAGTCACCTCAGAATCTCGTATG	60030	5.01369306977828	TruSeq Adapter, Index 1 (97% over 37bp)
GATCGGAAGAGCACACGTCTGAACTCCAGTCACCTCAGAATCTCGTATGC	42955	3.5875926338884896	TruSeq Adapter, Index 1 (97% over 37bp)
CACACGTCTGAACTCCAGTCACCTCAGAATCTCGTATGCCGTCTTCTGCT	3574	0.29849973398946483	RNA PCR Primer, Index 40 (100% over 41bp)
CAGATCGGAAGAGCACACGTCTGAACTCCAGTCACCTCAGAATCTCGTAT	2519	0.2103863542024236	TruSeq Adapter, Index 1 (97% over 37bp)
GAGATCGGAAGAGCACACGTCTGAACTCCAGTCACCTCAGAATCTCGTAT	1251	0.10448325887543942	TruSeq Adapter, Index 1 (97% over 37bp)

Overrepresented sequences

- Here < 1% of sequences contain adapters
  - trimming optional

Sequence	Count	Percentage	Possible Source
AACGACTCTCGGCAACGGATATCTCGGCTCTCGCATCGATGAAGAACGTA	102020	1.0707851766890004	No Hit
AATTCTAGAGCTAATACGTGCAACAAACCCCGACTTATGGAAGGGACGCA	89437	0.9387160737848865	No Hit
AAAGGATTGGCTCTGAGGGCTGGGCTCGGGGGTCCCAGTTCCGAACCCGT	89427	0.9386111154260659	No Hit
TACCTGGTTGATCCTGCCAGTAGTCATATGCTTGTCTCAAAGATTAAGCC	87604	0.9194772066130483	No Hit
ATTGGCTCTGAGGGCTGGGCTCGGGGGTCCCAGTTCCGAACCCGTCGGCT	65829	0.6909303802809273	No Hit
TCTAGAGCTAATACGTGCAACAAACCCCGACTTATGGAAGGGACGCATTT	65212	0.6844544495416888	No Hit
TAAAACGACTCTCGGCAACGGATATCTCGGCTCTCGCATCGATGAAGAAC	61582	0.646354565289767	No Hit
CTCGGATAACCGTAGTAATTCTAGAGCTAATACGTGCAACAAACCCCGAC	59180	0.6211435675010296	No Hit
ATGGATCCGTAACTTCGGGAAAAGGATTGGCTCTGAGGGCTGGGCTCGGG	56982	0.598073720232235	No Hit
AAAACGACTCTCGGCAACGGATATCTCGGCTCTCGCATCGATGAAGAACG	54813	0.5753082522040206	No Hit
CTAGAGCTAATACGTGCAACAAACCCCGACTTATGGAAGGGACGCATTTA	40019	0.4200328561646452	No Hit
AGAACTCCGCAGTTAAGCGTGCTTGGGCGAGAGTAGTACTAGGATGGGTG	39753	0.4172409638200141	No Hit
ACTCGGATAACCGTAGTAATTCTAGAGCTAATACGTGCAACAAACCCCGA	38867	0.4079416532284981	No Hit
ACGACTCTCGGCAACGGATATCTCGGCTCTCGCATCGATGAAGAACGTAG	38438	0.40343893963508914	No Hit
ACTTCGGGAAAAGGATTGGCTCTGAGGGCTGGGCTCGGGGGTCCCAGTTC	37406	0.3926072370047907	No Hit
AGATCGGAAGAGCACACGTCTGAACTCCAGTCACTGACCAATCTCGTATG	34199	0.35894709133098535	TruSeq Adapter, Index 4 (100% over 49bp)
GAACCTTGGGATGGGTCGGCCGGTCCGCCTTTGGTGTGCATTGGTCGGCT	34099	0.3578975077427782	No Hit

Overrepresented sequences

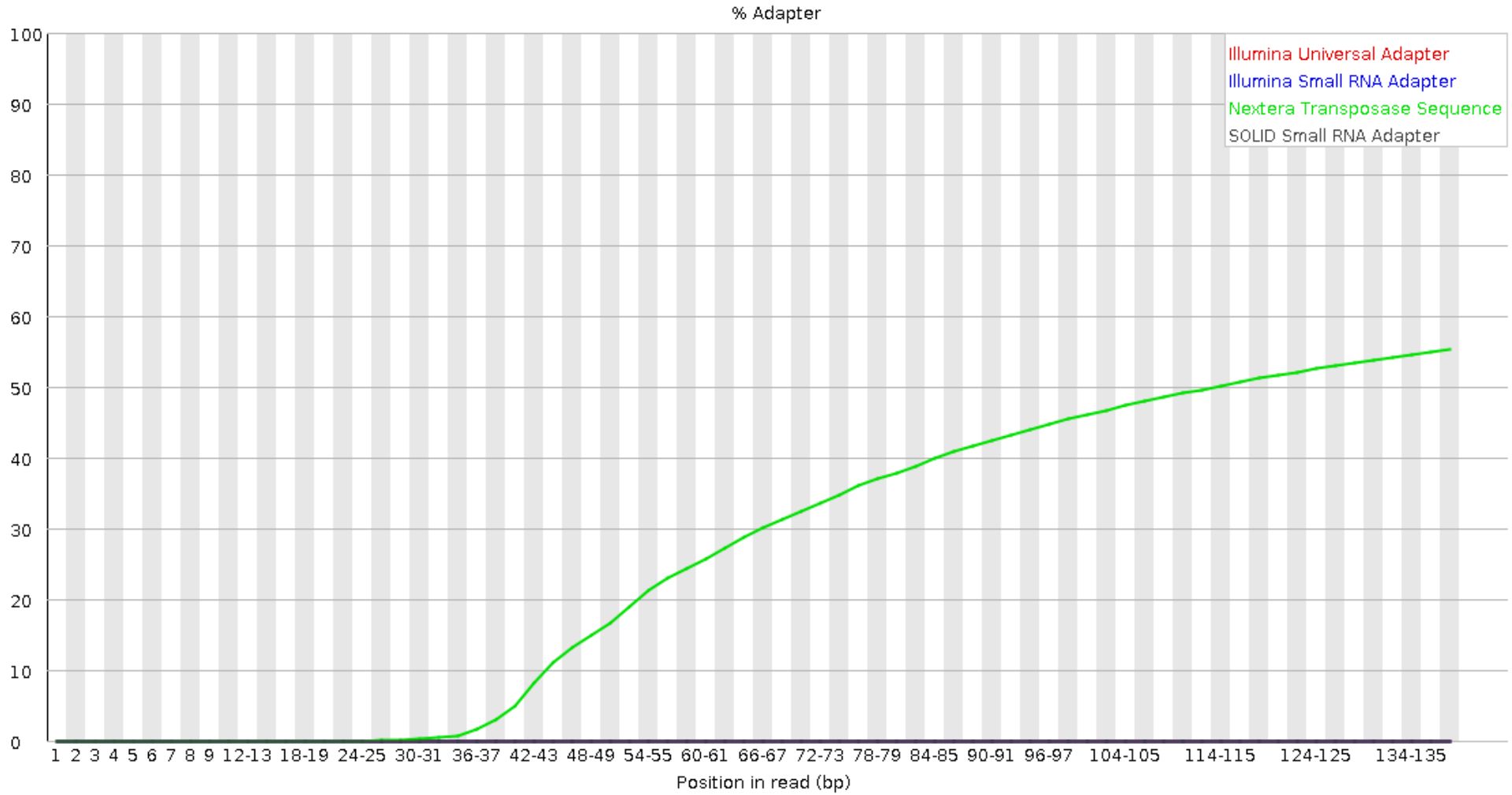


- Here nearly 1/3 of sequences some type of non-adapter contamination
  - **BLAST** the sequence to identify it

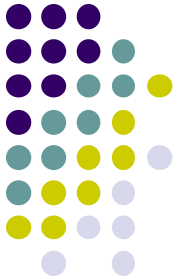
Sequence	Count	Percentage	Possible Source
GAAGGTCACGGCGAGACGAGCCGTTTATCATTACGATAGGTGTCAAGTGG	5632816	32.03026785752871	No Hit
TATTCTGGTGTCCTAGGCGTAGAGGAACAACACCAATCCATCCCGAACTT	494014	2.8091456822607364	No Hit
TCAAACGAGGAAAGGCTTACGGTGGATACCTAGGCACCCAGAGACGAGGA	446641	2.539765344040083	No Hit
TAAAACGACTCTCGGCAACGGATATCTCGGCTCTCGCATCGATGAAGAAC	179252	1.0192929387357474	No Hit
GAAGGTCACGGCGAGACGAGCCGTTTATCATTACGATAGGGGTCAAGTGG	171681	0.9762414422996221	No Hit
AACGACTCTCGGCAACGGATATCTCGGCTCTCGCATCGATGAAGAACGTA	143415	0.8155105483274229	No Hit
AGAACATGAAACCGTAAGCTCCCAAGCAGTGGGAGGAGCCCTGGGCTCTG	111584	0.6345077504066322	No Hit
AAAACGACTCTCGGCAACGGATATCTCGGCTCTCGCATCGATGAAGAACG	111255	0.6326369351474214	No Hit
ATTACGATAGGTGTCAAGTGGAAGTGCAGTGATGTATGCAGCTGAGGCAT	73682	0.41898300890326096	No Hit
GAAGGTCACGGCGAGACGAGCCGTTTATCATTACGATAGGTGTCAAGGGG	71661	0.4074908580252516	No Hit
GGATGCGATCATACCAGCACTAATGCACCGGATCCCATCAGAACTCCGCA	69548	0.3954755612388914	No Hit
ATATTCTGGTGTCCTAGGCGTAGAGGAACAACACCAATCCATCCCGAACT	54017	0.30716057099328803	No Hit

Adapter Content report

- Newer versions of FastQC have a separate Adapter Content report
  - provides a per-base % adapter trace (Transposon-seq below)

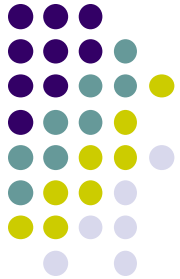


Dealing with 3’ adapters



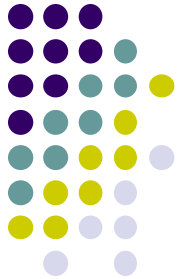
- Three main options:
  1. ***Hard trim*** all sequences by specific amount
  2. ***Remove*** adapters specifically
  3. Perform a ***local alignment*** (vs ***global***)

Hard trim by specific length



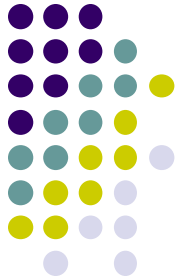
- E.g. trim 100 base reads to 50 bases
- ***Pro:***
  - Can eliminate vast majority of adapter contamination
  - Fast, easy to perform
  - Low quality 3’ bases also removed
- ***Con:***
  - Removes information you may want
    - e.g. splice junctions for RNA-seq, coverage for mutation analysis
  - Not suitable for very short library fragments
    - e.g. miRNA libraries

Trim adapters specifically



- ***Pro:***
  - Can eliminate vast majority of adapter contamination
  - Minimal loss of sequence information
    - still ambiguous: are 3'-most bases part of sequence or adapter?
- ***Con:***
  - Requires knowledge of insert fragment structure and adapters
  - Slower process; more complex to perform
  - Results in a heterogeneous pool of sequence lengths
    - can confuse some downstream tools (rare)
- Specific adapter trimming is most common for RNA-seq
  - most transcriptome-aware aligners need adapter-trimmed reads

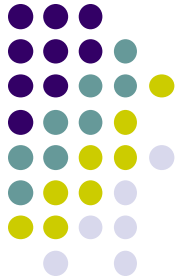
FASTQ trimming and adapter removal



- Tools:
  - **cutadapt** – https://cutadapt.readthedocs.io/en/stable/
  - **trimmomatic** – http://www.usadellab.org/cms/?page=trimmomatic
  - FASTX-Toolkit – http://hannonlab.cshl.edu/fastx_toolkit/

- Features:
  - hard-trim specific number of bases
  - trimming of low quality bases
  - specific trimming of adapters
  - support for trimming paired end read sets (except FASTX)
    - reads shorter than a specified length *after trimming* are typically discarded
    - leads to different sets of R1 and R2 reads unless care is taken
      - aligners do not like this!
  - **cutadapt** has protocol for separating reads based on internal barcode

Local vs. Global alignment



- ***Global*** alignment
  - requires query sequence to map ***fully*** (end-to-end) to reference
- ***Local*** alignment
  - allows a ***subset*** of the query sequence to map to reference
    - "untemplated" adapter sequences will be "soft clipped" (ignored)

global *(end-to-end) alignment of query*

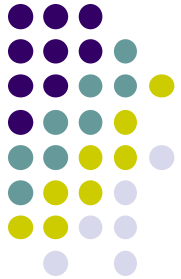
local *(subsequence) alignment of query*

CACAAGTACAATTATACAC **CTAGCTTATCGCCCTGAAGGACT**

TACATACACAAGTACAATTATACACAGACATTAGTTCTTATCGCCCTGAAAATTCTCC

reference sequence

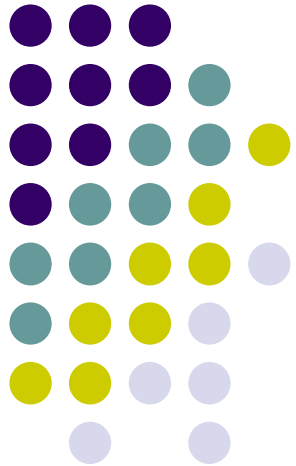
Perform local alignment

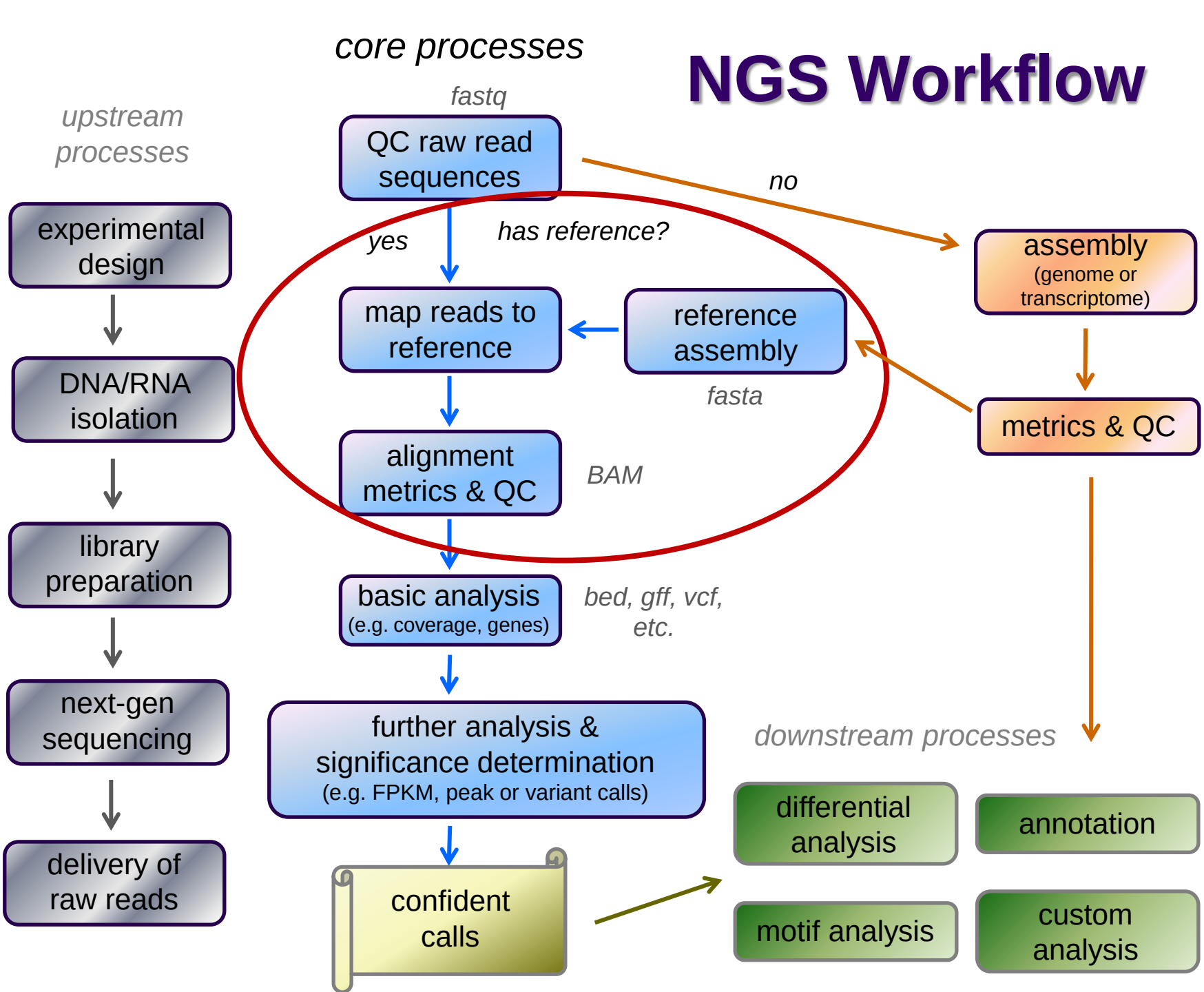


- ***Pro:***
  - mitigates adapter contamination while retaining full query sequence
  - minimal ambiguity
    - still a bit ambiguous: are 3'-most bases part of sequence or adapter?
- ***Con:***
  - not supported by many aligners
    - e.g. not by the **hisat2** or **tophat** splice-aware aligners for RNAseq
    - ***Tip***: the **STAR** RNAseq aligner can perform adapter *trimming* as part of alignment
  - slower alignment process
  - more complex post-alignment processing may be required
- Aligners with local alignment support:
  - **bwa mem**
  - **bowtie2 --local**

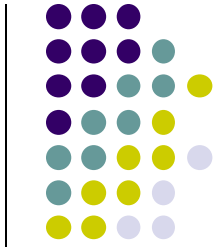
Part 4: Alignment to a reference assembly

- Alignment overview & concepts
- Preparing a reference genome
- Alignment workflow steps



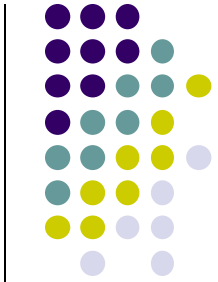

core processes
NGS Workflow
upstream processes
fastq
QC raw read sequences
no
experimental design
yes
has reference?
assembly
(genome or transcriptome)
map reads to reference
reference assembly
fasta
DNA/RNA isolation
metrics & QC
alignment metrics & QC
BAM
library preparation
basic analysis
(e.g. coverage, genes)
bed, gff, vcf, etc.
next-gen sequencing
further analysis & significance determination
(e.g. FPKM, peak or variant calls)
downstream processes
differential analysis
annotation
delivery of raw reads
confident calls
motif analysis
custom analysis

Short Read Aligners



- Short read mappers determine placement of ***query sequences*** (your reads) against a known ***reference***
  - **BLAST**:
    - ***one*** query sequence (or a few)
    - want many matches for each
  - short read aligners
    - many ***millions*** of query sequences
    - want only one "best" mapping (or a few) for each
- Many aligners available! Two of the most popular
  - **bwa** (Burrows Wheeler Aligner) by Heng Li
    http://bio-bwa.sourceforge.net/
  - **bowtie2** – part of the Johns Hopkins "Tuxedo" suite of tools
    http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml
  - Given similar input parameters, they produce similar alignments
    - and both run relatively quickly

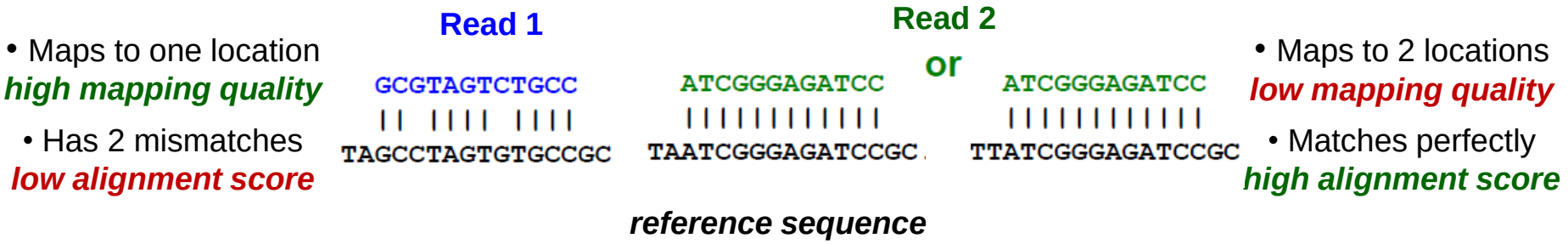
Aligner criteria



- ***Adoption and currency***
  - ***widespread use by bioinformatics community***
  - ***still being actively developed***
- Features
  - well understood algorithm(s)
  - support for a variety of input formats and read lengths
  - detection of insertions/deletions (indels) and gaps
  - makes use of base qualities
  - handling of multiple matches
- Usability
  - configurability and transparency of options
  - ease of installation and use
- Resource requirements
  - speed (“fast enough”)
  - scalability (takes advantage of multiple processors)
  - reasonable memory footprint

Mapping *vs* Alignment

- ***Mapping*** determines one or more ***positions*** (a.k.a. ***seeds*** or ***hits***) where a read shares a *short* sequence with the reference
- ***Alignment*** starts with the seed and determines how read bases are best ***matched***, base-by-base, around the seed
- Mapping quality and alignment scores are both reported
  - High mapping quality ≠ High alignment score
  - ***mapping quality*** describes ***positioning***
    - reflects the probability that the read is ***incorrectly*** mapped to the reported location
    - is a Phred score: $P(\text{incorrectly mapped}) = 10^{-\text{mappingQuality}/10}$
      - also reflects the ***complexity*** or information content of the sequence (***mappability***)
  - ***alignment score*** describes ***fit***
    - reflects the correspondence between the read and the reference sequence



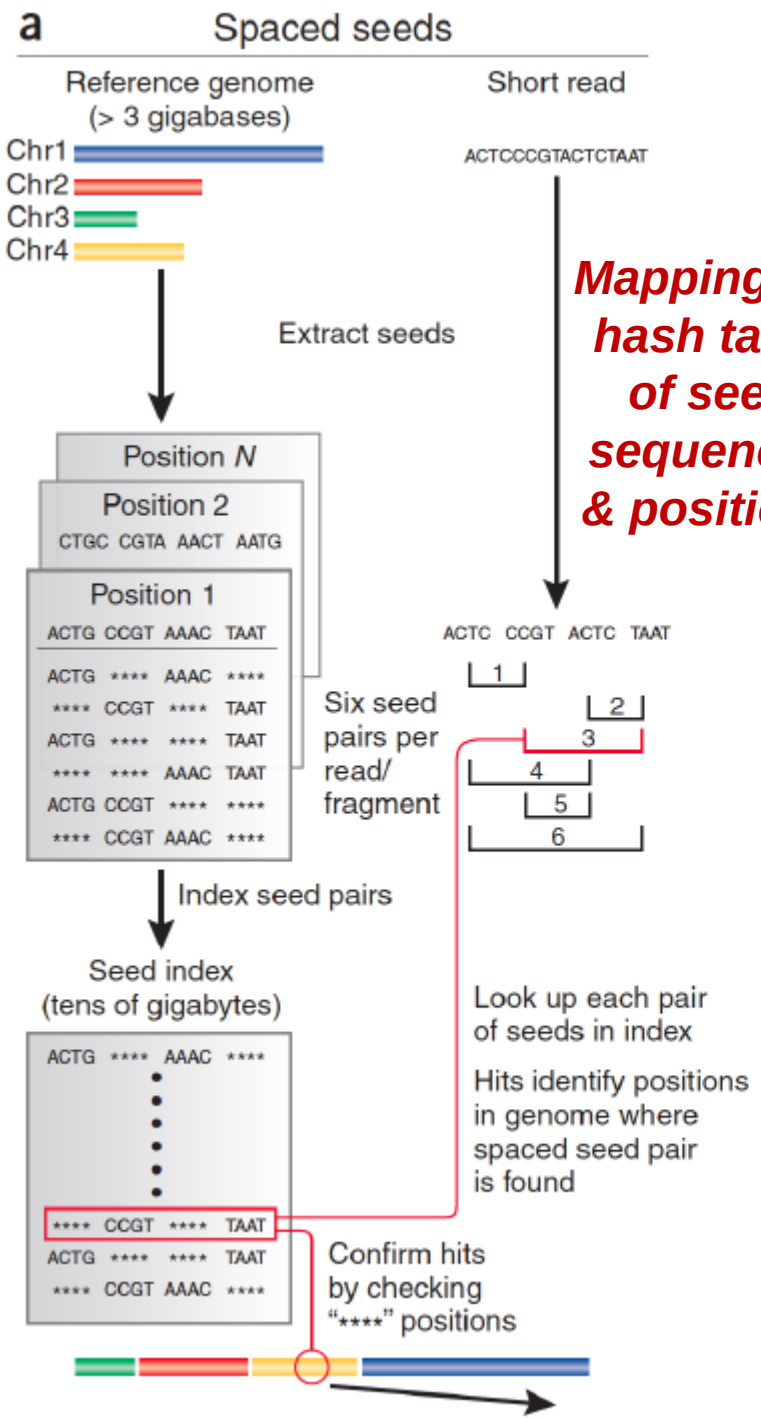
Mapping algorithms

Two main mapping algorithms: *spaced seeds*, *suffix-array tries*

	Algorithm	Gapped	Quality-aware	Colorspace aware
BLAST	Hash table	Y	N	N
BLAT/SSHA2	Hash table	N	N	N
MAQ	Spaced seed	N	N	N
RMAP	Spaced seed	N	Y	N
ZOOM	Spaced seed	N	--	N
SOAP	Spaced seed	N	N	N
Eland	Spaced seed	N	N	N
SHRiMP	Q-gram/multi-seed	Y	Y	Y
BFAST	Q-gram/multi-seed	Y	Y	Y
Novoalign	Multi-seed + Vectorized SW	Y	Y	Y
clcBio	Multi-seed + Vectorized SW	Y	Y	Y
MUMmer	Tries	Y	N	N
OASIS	Tries	Y	--	--
VMATCH	Tries	Y	--	--
BWA/BWA-SW	Tries	Y	Y	Y
BOWTIE	Tries	Y	Y	Y
SOAP2	Tries	Y	N	N
Saruman	Exact (GPU)	Y	--	N

courtesy of Matt Vaughn, TACC

trie = tree structure for fast text retrieval.



Mapping via hash table of seed sequences & positions

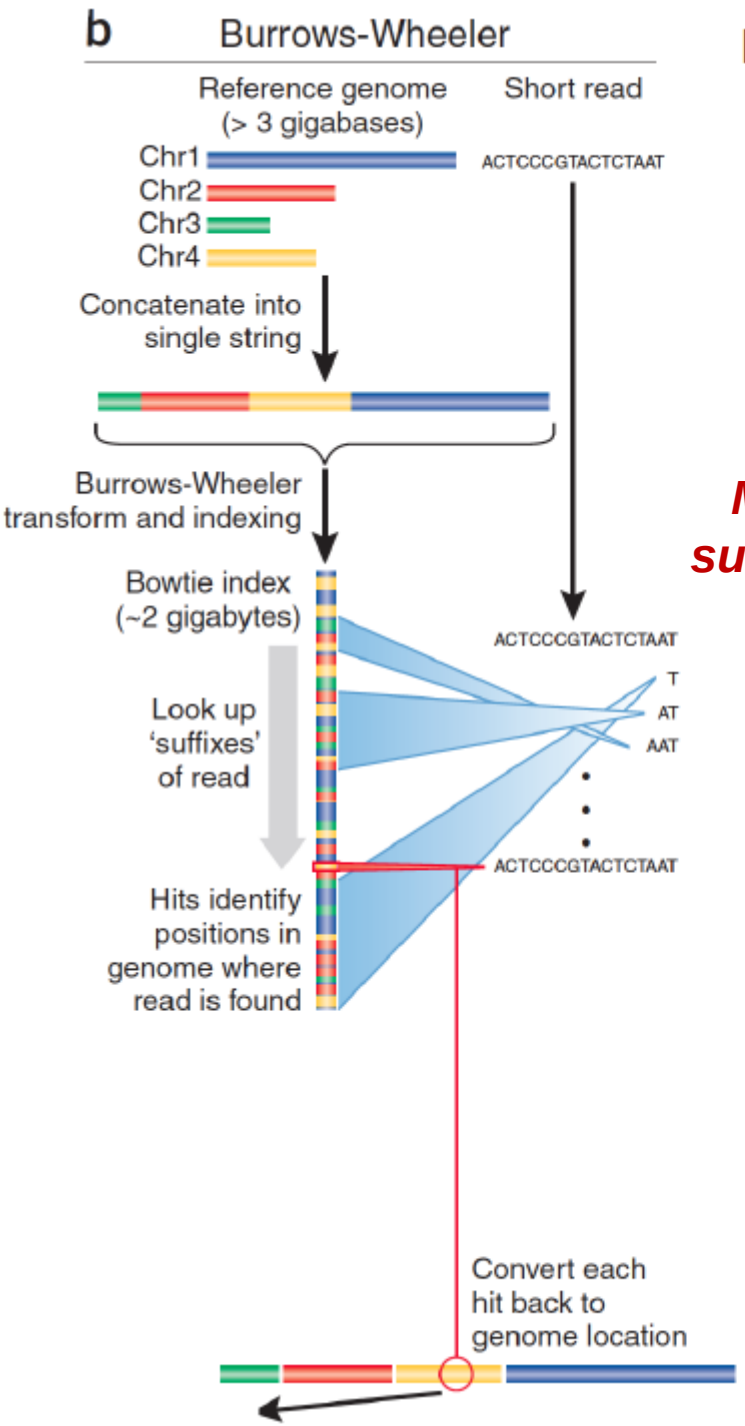
Hash table enables lookup of exact matches.

Subsequence	Reference Positions
ATAGCTAATCCAAA	2341, 2617264
ATAGCTAATCCAAT	
ATAGCTAATCCAAC	134, 13311, 732661,
ATAGCTATCCAAAG	
ATAGCTAATCCATA	
ATAGCTAATCCATT	3452
ATAGCTAATCCATC	
ATAGCTATCCAATG	234456673

Table is sorted and complete so you can jump immediately to matches. (But this can take a lot of memory.)

May include N bases, skip positions, etc.

Trapnell, C. & Salzberg, S. L. How to map billions of short reads onto genomes. *Nature Biotech.* **27**, 455–457 (2009).

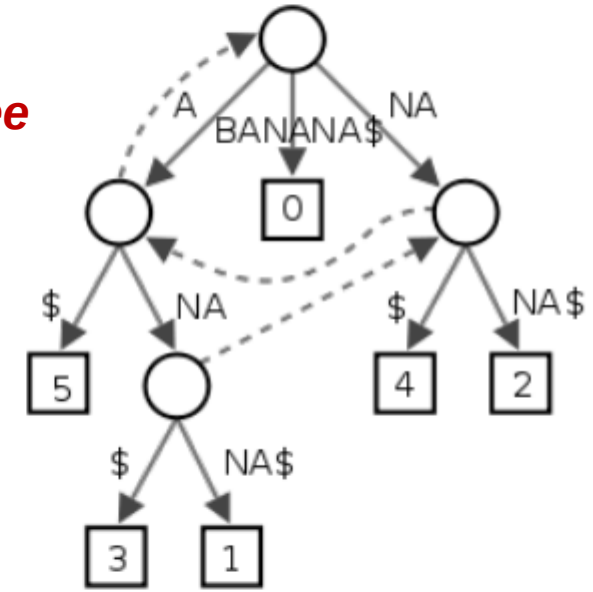


Burrows-Wheeler transform compresses sequence.

Input	SIX.MIXED.PIXIES.SIFT.SIXTY.PIXIE.DUST.BOXES
Output	TEXYDST.E.IXIXIXXSSMPPS.B..E.S.EUSFXDIIOIIIT

Suffix tree enables fast lookup of subsequences.

Mapping via suffix array tree



http://en.wikipedia.org/wiki/Suffix_tree

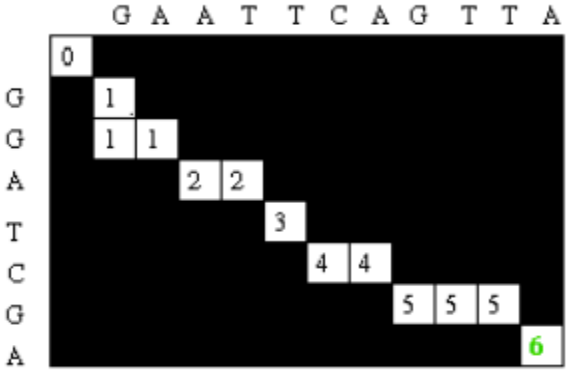
Exact matches at all positions below a node.

Trapnell, C. & Salzberg, S. L. How to map billions of short reads onto genomes. *Nature Biotech.* **27**, 455–457 (2009).

Alignment via dynamic programming

- Dynamic programming algorithm (Smith-Waterman | Needleman-Wunsch)

		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1	1
G	0	1	1	1	1	1	1	1	2	2	2	2
A	0	1	1	2	2	2	2	2	2	2	2	3
T	0	1	2	2	3	3	3	3	3	3	3	3
C	0	1	2	2	3	3	4	4	4	4	4	4
G	0	1	2	2	3	3	4	4	5	5	5	5
A	0	1	2	3	3	3	4	5	5	5	5	6



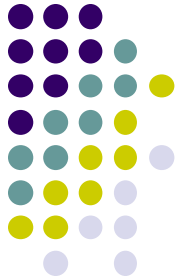

```
G _ A A T T C A G T T A
|   |   | |   |     |
G G _ A _ T C _ G _ _ A
```

- **Alignment score = Σ**
  - match reward
  - base mismatch penalty
  - gap open penalty
  - gap extension penalty
  - rewards and penalties may be adjusted for quality scores of bases involved

Reference sequence

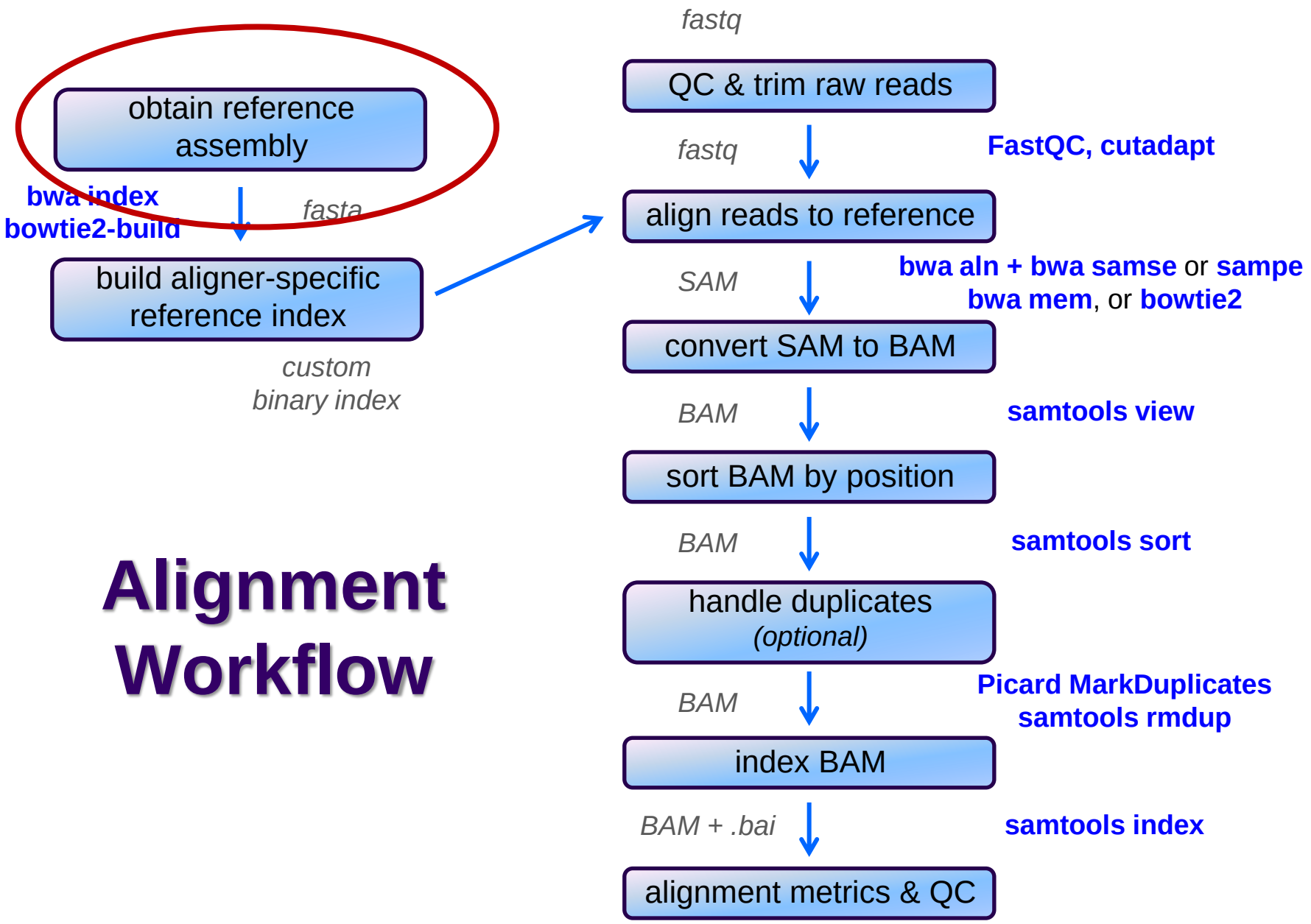
```
ATTTGCGATCGGATGAAGACGAA
|||||||||||||||
ATTTGCGATCGGATGTTGACTTT

ATTTGCGATCGGATGAAGACG..AA
|||||||||||||||XX|||Xii||
ATTTGCGATCGGATGTTGACTTTAA
```

Paired End mapping



- Having paired-end reads improves mapping
  - mapping one read with high confidence anchors the pair
    - even when the mate by itself maps several places equally
- Three possible outcomes of mapping an R1/R2 pair
  1. only one of a pair might map ***(singleton/orphan)***
  2. both reads can map within the most likely distance range and with correct orientation ***(proper pair)***
  3. both reads can map but with an unexpected insert size or orientation, or to different contigs ***(discordant pair)***
- Insert size is reported in the alignment record
  - for both proper and discordant pairs
    - but insert size is only meaningful for proper pairs

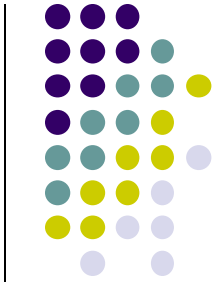
Alignment Workflow



http://bio-bwa.sourceforge.net/bwa.shtml

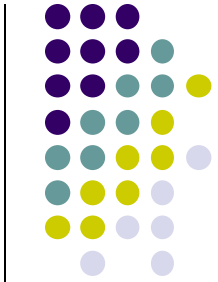
http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml

Obtaining a reference



- Assembled genomes
  - Ensembl, UCSC, Gencode for eukaryotes
    - FASTA files (**.fa, .fasta**)
    - annotations (genome feature files, **.gtf .gff .gff3**)
  - NCBI RefSeq (or GenBank) for prokaryotes/microbes (prefer RefSeq)
    - Can obtain both FASTA sequences and annotations
  - For species without a good assembly, the assembly of a closely related species is often used
- A reference is just a set of sequences of interest
  - *any set of named DNA sequences*
    - e.g. chromosomes (partial or complete), technically referred to as ***contigs***
  - a transcriptome (set of transcribed gene sequences for an organism)
  - miRNA hairpin sequences from miRBase
  - rRNA/tRNA genes (e.g. for filtering)
  - one or more amplicons or plasmid constructs

FASTA format



- FASTA files contain a set of sequence records
  - can be DNA, RNA, protein sequences
  - ***sequence name*** line
    - ***always*** starts with **>**
      - followed by a ***name*** and other (optional) descriptive information
  - one or more line(s) of ***sequence characters***
    - ***never*** starts with **>**
- Mitochondrial chromosome sequence, human from UCSC hg19

```
>chrM
GATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTCTCCATGCAT
TTGGTATTTTCGTCTGGGGGGTGTGCACGCGATAGCATTGCGAGACGCTG
GAGCCGGAGCACCCTATGTCGCAGTATCTGTCTTTGATTCCTGCCTCATT ...
```

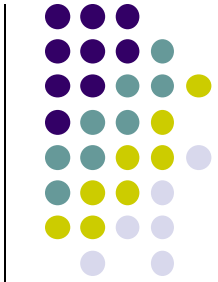
- Let-7e miRNA, human from miRBase v21

```
>hsa-let-7e MI0000066 Homo sapiens let-7e stem-loop
CCCGGGCUGAGGUAGGAGGUUGUAUAGUUGAGGAGGACACCCAAGGAGAUCACUAUACGG
CCUCCUAGCUUUCCCCAGG
```

- P53 protein, from UniProt

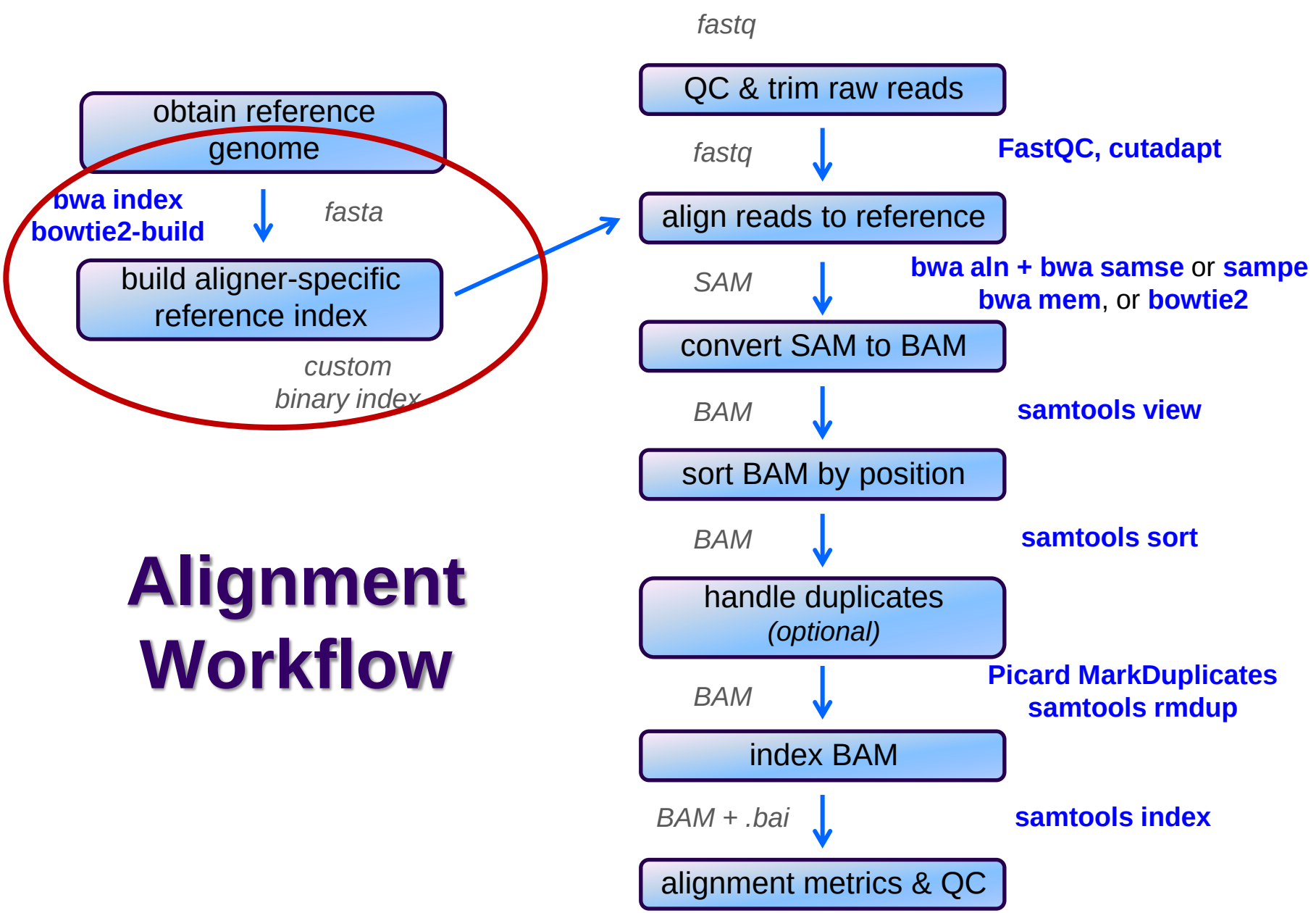
```
>sp|P04637|P53_HUMAN Cellular tumor antigen p53 OS=Homo sapiens GN=TP53
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGP
DEAPRMPEAAPPVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAK ...
```

Reference considerations



- Is it appropriate to your study?
  - close enough to your species? complete?
- From which source? And which version?
  - UCSC hg19 vs Ensembl GRCh37
- What annotations exist?
  - references lacking feature annotations are much more challenging
- Does it contain repeats?
  - if so, are they masked in your FASTA?
- Watch out for sequence name issues!
  - sequence names may be different between UCSC/Ensembl
    - e.g. “chr12” vs “12”
  - ***annotation sequence names must match names in your reference!***
  - very long sequence names can cause problems
    - rename: `>hsa-let-7e_MI0000066_Homo_sapiens_let-7e stem-loop`
    - to: `>hsa-let-7e`

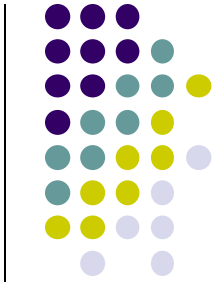
Alignment Workflow



http://bio-bwa.sourceforge.net/bwa.shtml

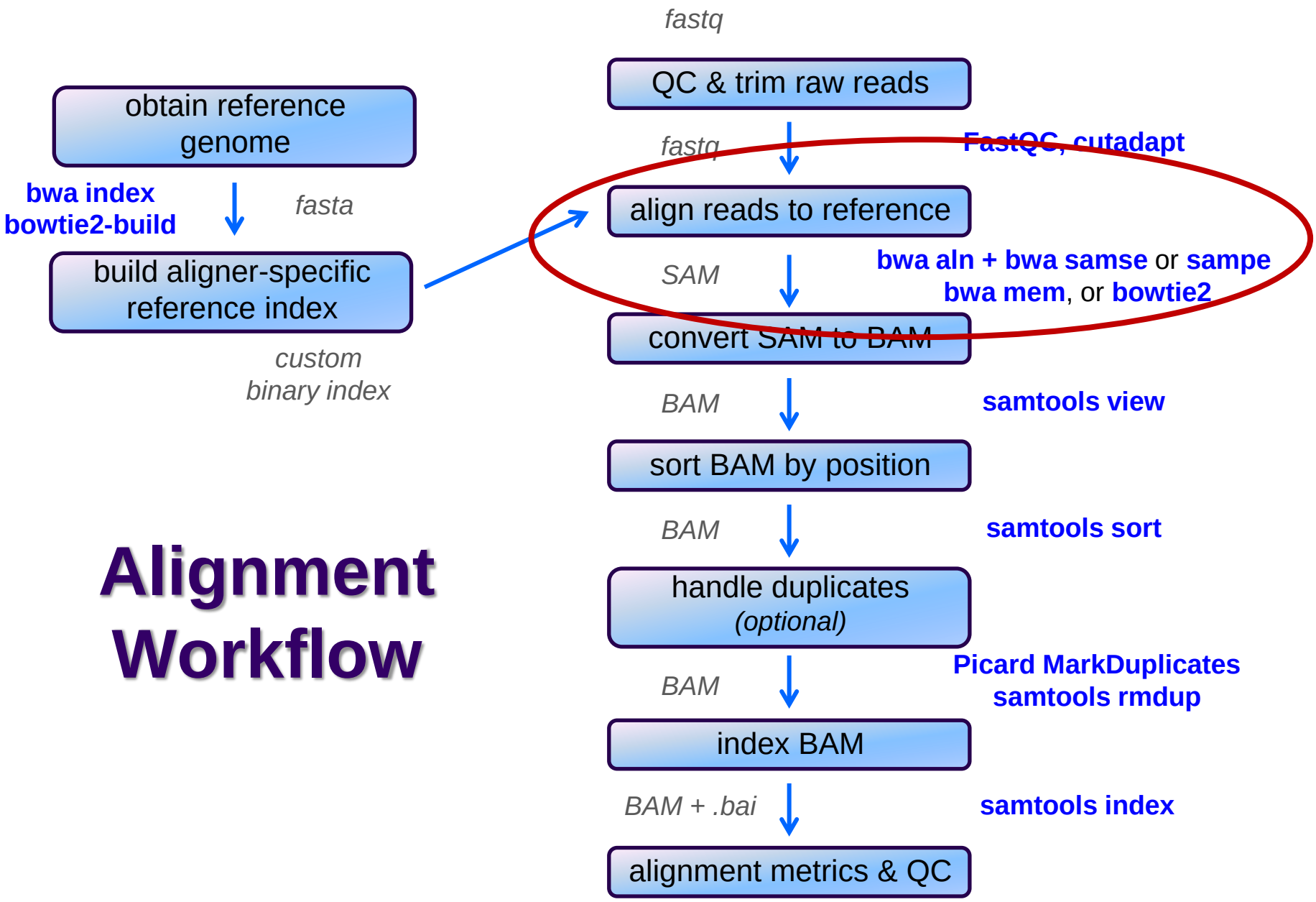
http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml

Building a reference index



- Index format is *specific to each aligner*
  - may take several hours to build
    - but you build each index once, then use for multiple alignments
  - Input:
    - one or more FASTA files containing DNA sequences
      - i.e. convert RNA sequences with U's to cDNA sequences with T's
    - annotations (genome feature files, **.gtf**) are sometimes also used to build a transcriptome-aware index for RNA-seq (e.g. STAR aligner)
      - but annotations will ***definitely*** be needed for downstream analysis
  - Output:
    - a number of binary files the aligner will use
- Best practice:
  - build each index in its own appropriately named directory, e.g.
    - **refs/bowtie2/UCSC/hg38**
    - **refs/bwa/Ensembl/GRCh38**

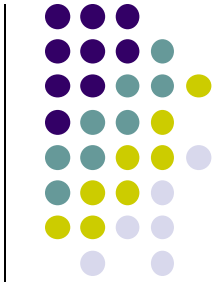
Alignment Workflow



http://bio-bwa.sourceforge.net/bwa.shtml

http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml

SAM file format



- Aligners take FASTQ as input, output alignments in **S**equence **A**lignment **M**ap (SAM) format
  - community file format that describes how reads map (and align) to a reference
  - the Bible: http://samtools.github.io/hts-specs/SAMv1.pdf
    - and now https://github.com/samtools/hts-specs/blob/master/SAMtags.pdf
- SAM file consists of
  - a ***header***
    - includes reference sequence names and lengths
  - ***alignment records***, one for each sequence read
    - can include both mapped and unmapped reads
    - alignments for R1 and R2 reads have ***separate records***
      - with fields that refer to the mate
    - records have11 fixed fields + extensible-format **key:type:value** tuples

SAM file format
Fixed fields (tab-separated)

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	[!-?A-~]{1,255}	Query template NAME
2	FLAG	Int	[0,2^{16}-1]	bitwise FLAGs
3	RNAME	String	*|[!-()+-<>-~][!-~]*	Reference sequence NAME
4	POS	Int	[0,2^{29}-1]	1-based leftmost mapping POSition
5	MAPQ	Int	[0,2^{8}-1]	MAPping Quality
6	CIGAR	String	*|([0-9]+[MIDNSHPX=])+	CIGAR string
7	RNEXT	String	*|=|[!-()+-<>-~][!-~]*	Ref. name of the mate/next segment
8	PNEXT	Int	[0,2^{29}-1]	Position of the mate/next segment
9	TLEN	Int	[-2^{29}+1,2^{29}-1]	observed Template LENgth
10	SEQ	String	*|[A-Za-z=.]+	segment SEQuence
11	QUAL	String	[!-~]+	ASCII of Phred-scaled base QUALity+33

read name from fastq

contig + start = locus

use this to find end coordinate

insert size, if paired

```
SRR030257.264529   99   NC_012967   1521   29   34M2S   =   1564   79
CTGGCCATTATCTCGGTGGTAGGACATGGCATGCCC
AAAAAA;AA;AAAAAA??A%.;?&'3735',()0*,
XT:A:M  NM:i:3  SM:i:29  AM:i:29  XM:i:3  XO:i:0  XG:i:0  MD:Z:23T0G4T4
```

positive for plus strand reads

```
SRR030257.2669090   147   NC_012967   1521   60   36M   =   1458   -99
CTGGCCATTATCTCGGTGGTAGGTGATGGTATGCGC
<<9:<<AAAAAAAAAAAAAAAAAAAAAAAAAAAA
XT:A:U  NM:i:0  SM:i:37  AM:i:37  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36
```

negative for minus strand reads

Sometimes a CIGAR is just a way of describing how a read is aligned…

```
Ref   CTGGCCATTATCTC--GGTGGTAGGACATGGCATGCCC
Read        aaATGTCGCGGTG.TAGGAggatcc
```

2S5M2I4M1D5M6S

Op	BAM	Description
M	0	alignment match (can be a sequence match or mismatch)
I	1	insertion to the reference
D	2	deletion from the reference
N	3	skipped region from the reference
S	4	soft clipping (clipped sequences present in SEQ)
* H	5	hard clipping (clipped sequences NOT present in SEQ)
* P	6	padding (silent deletion from padded reference)
* =	7	sequence match
* X	8	sequence mismatch

"N" indicates splicing event in RNA-seq BAMs

*Rarer / newer

CIGAR = "Concise Idiosyncratic Gapped Alignment Report"

SAM format – Bitwise flags

https://wikis.utexas.edu/display/CoreNGSTools/Decimal+and+Hexadecimal

Bit Decimal	Hex	Description	
1	0x1	template having multiple segments in sequencing	*1 = part of a read pair*
2	0x2	each segment properly aligned according to the aligner	*1 = "properly" paired*
4	0x4	segment unmapped	*read did map = 0* *1 = read did not map*
8	0x8	next segment in the template unmapped	*1 = mate did not map*
16	0x10	SEQ being reverse complemented	*plus strand read = 0* *1 = minus strand read*
32	0x20	SEQ of the next segment in the template being reverse complemented	*1 = mate on minus strand*
64	0x40	the first segment in the template	*1 = R1 read*
128	0x80	the last segment in the template	*1 = R2 read*
256	0x100	secondary alignment	*1 = secondary alignment*
512	0x200	not passing filters, such as platform/vendor quality controls	
1024	0x400	PCR or optical duplicate	*1 = marked as duplicate*
2048	0x800	supplementary alignment	*1 = maps to ALT contig*

```
SRR030257.264529    99   NC_012967   1521    29   34M2S   =   1564    79
    CTGGCCATTATCTCGGTGGTAGGACATGGCATGCCC
    AAAAAA;AA;AAAAAA??A%.;?&'3735',()0*,
    XT:A:M  NM:i:3  SM:i:29  AM:i:29  XM:i:3  XO:i:0  XG:i:0  MD:Z:23T0G4T4

SRR030257.2669090   147  NC_012967   1521    60   36M     =   1458    -99
    CTGGCCATTATCTCGGTGGTAGGTGATGGTATGCGC
    <<9:<<AAAAAAAAAAAAAAAAAAAAAAAAAAAA
    XT:A:U  NM:i:0  SM:i:37  AM:i:37  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36
```

Decimal	Hex
99	= 0x63
= 64	= 0x40
+ 32	+ 0x20
+ 2	+ 0x02
+ 1	+ 0x01
147	= 0x93
= 128	= 0x80
+ 16	+ 0x10
+ 2	+ 0x02
+ 1	+ 0x01

http://broadinstitute.github.io/picard/explain-flags.html

SAM file format
key:type:value tuples

https://github.com/samtools/hts-specs/blob/master/SAMtags.pdf

Tag[1]	Type	Description
X?	?	Reserved fields for end users (together with Y? and Z?)

***alignment detail**: describes alignment of query to reference*

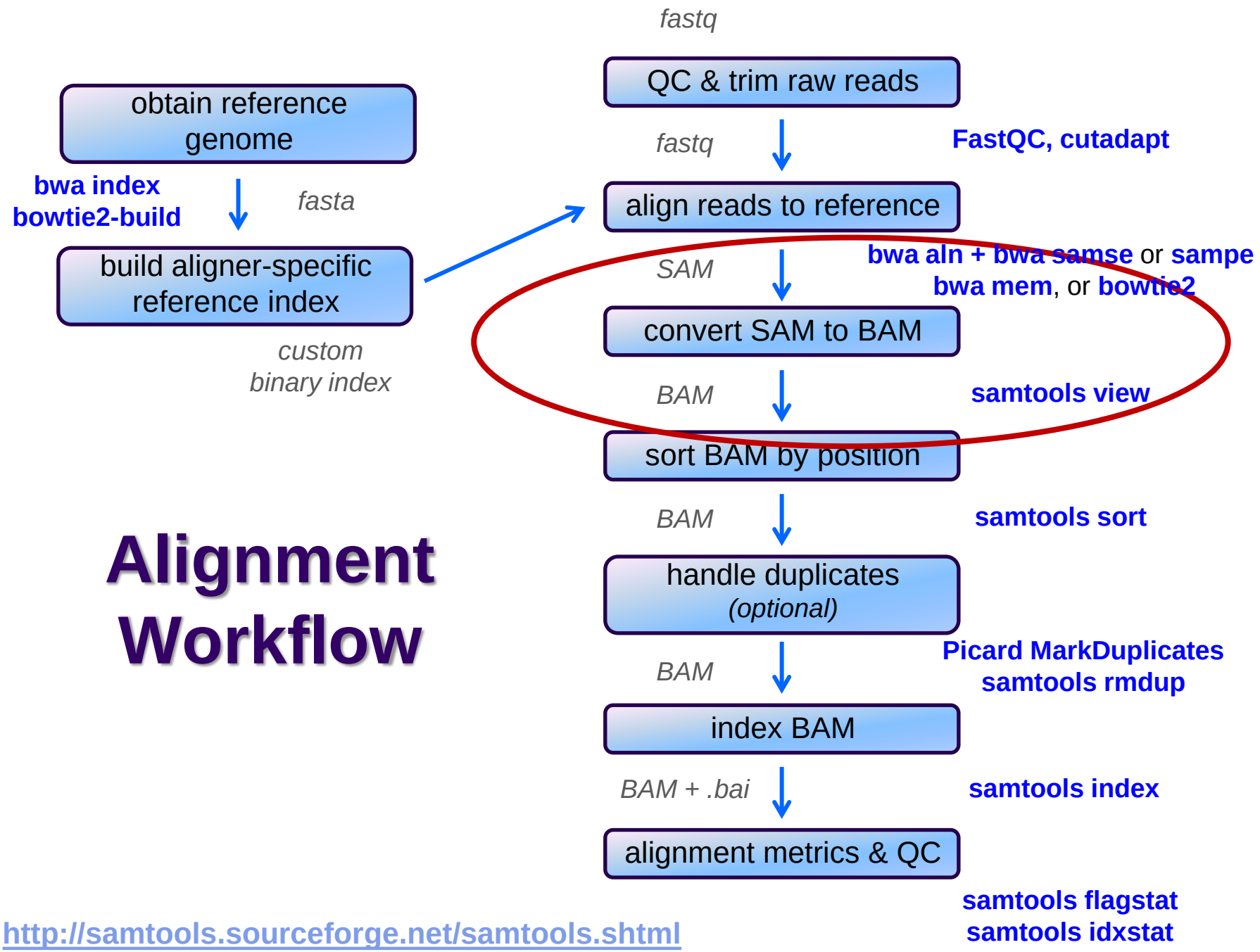
MD	Z	String for mismatching positions. *Regex*: [0-9]+(([A-Z]|\^[A-Z]+)[0-9]+)*[2]
MQ	i	Mapping quality of the mate/next segment
NH	i	Number of reported alignments that contains the query in the current record
NM	i	Edit distance to the reference, including ambiguous bases but excluding clipping

***edit distance** = # mismatches + insertions + deletions*

[2]The MD field aims to achieve SNP/indel calling without looking at the reference. For example, a string '10A5^AC6' means from the leftmost reference base in the alignment, there are 10 matches followed by an A on the reference which is different from the aligned read base; the next 5 reference bases are matches followed by a 2bp deletion from the reference; the deleted sequence is AC; the last 6 bases are matches. The MD field ought to match the CIGAR string.

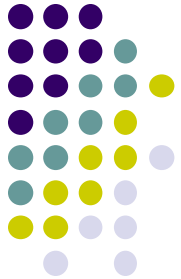
```
SRR030257.264529    99   NC_012967   1521   29   34M2S   =   1564   79
    CTGGCCATTATCTCGGTGGTAGGACATGGCATGCCC
    AAAAAA:AA;AAAAAA??A%.;?&'3735',()0*,
    XT:A:M  NM:i:3  SM:i:29  AM:i:29  XM:i:3  XO:i:0  XG:i:0  MD:Z:23T0G4T4
```

Alignment Workflow

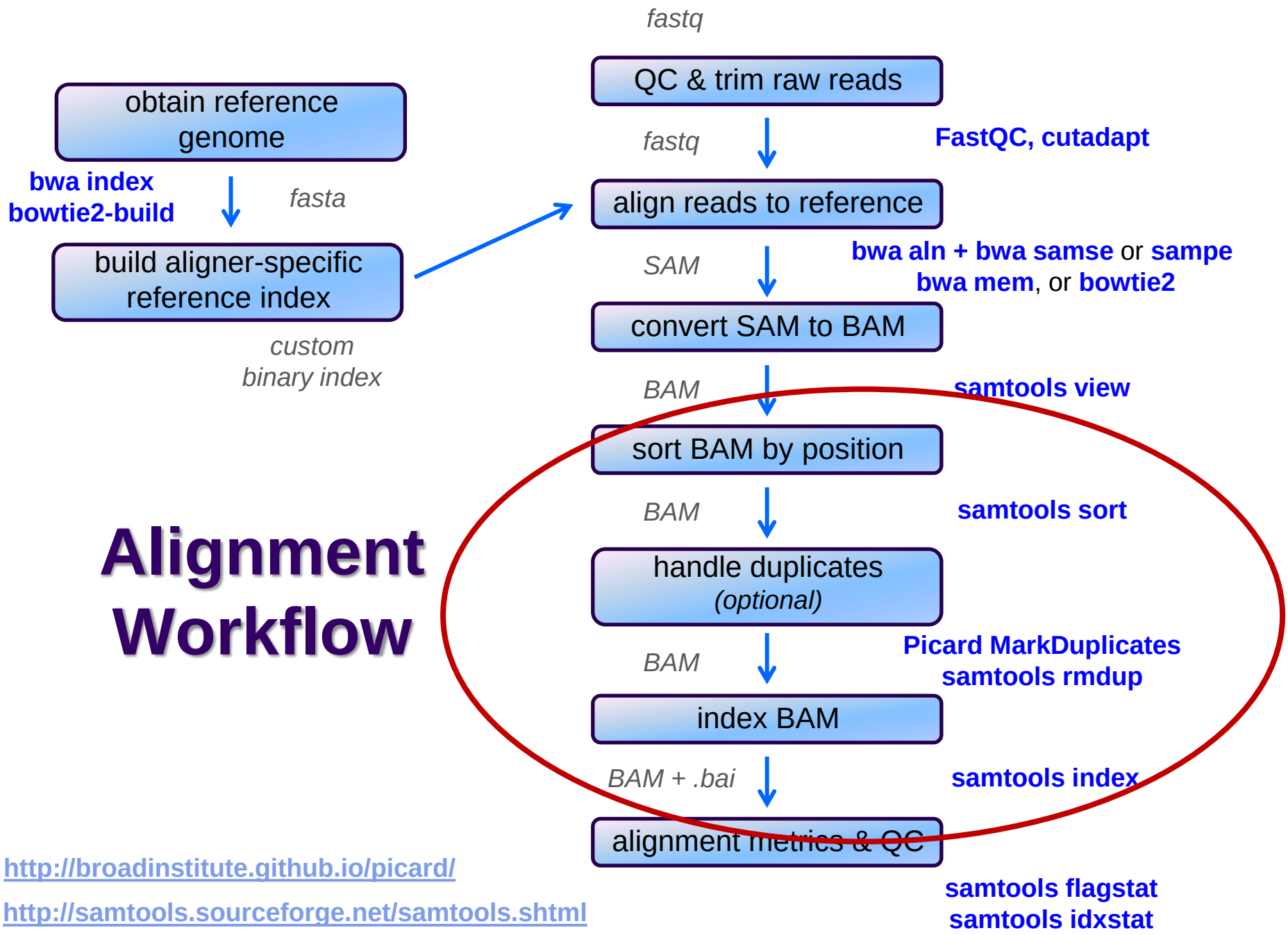


http://samtools.sourceforge.net/samtools.shtml

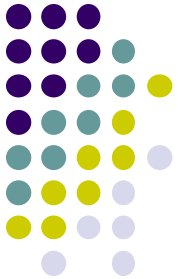
SAM / BAM files



- SAM and BAM are two forms of the same data
  - SAM – **S**equence **A**lignment **M**ap
    - plain text format
  - BAM – **B**inary **A**lignment **M**ap
    - ***same data*** in a custom compressed (**gzip**'d) format
- Differences
  - BAMs are ***much*** smaller than SAM files due to compression
  - BAM files support fast random access; SAM files do not
    - requires the BAM file to be *indexed*
  - most tools support BAM format and may require indexing
- Best practices
  - remove intermediate SAM and BAM files created during alignment and only save the final sorted, indexed BAM
  - keep your alignment artifacts (BAM, statistics files, log files) separate from the original FASTQ files
    - alignments can be re-generated – raw sequences cannot

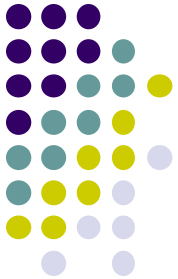

fastq
QC & trim raw reads
fastq
FastQC, cutadapt
obtain reference genome
bwa index
bowtie2-build
fasta
align reads to reference
build aligner-specific reference index
SAM
bwa aln + bwa samse or sampe
bwa mem, or bowtie2
convert SAM to BAM
custom binary index
BAM
samtools view
sort BAM by position
BAM
samtools sort
handle duplicates (optional)
BAM
Picard MarkDuplicates
samtools rmdup
index BAM
BAM + .bai
samtools index
alignment metrics & QC
samtools flagstat
samtools idxstat
Alignment Workflow
http://broadinstitute.github.io/picard/
http://samtools.sourceforge.net/samtools.shtml

Sorting / indexing BAM files



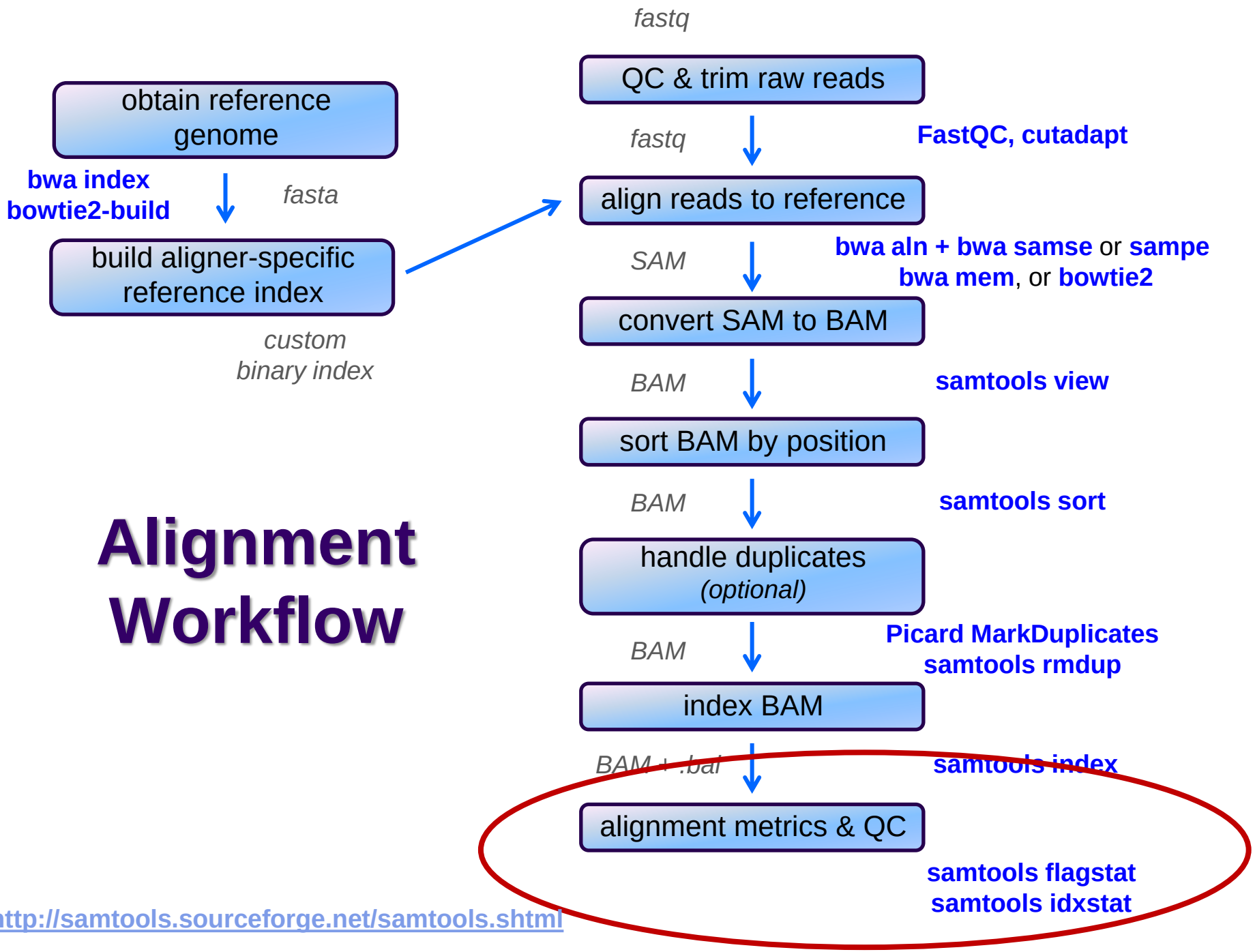
- SAM created by aligner contains read records in ***name order***
  - same order as read names in the input FASTQ file
    - R1, R2 have ***adjacent*** SAM records
  - SAM → BAM conversion does not change the name-sorted order
- Sorting BAM puts records in ***position (locus) order***
  - by contig name then leftmost start position
    - contig name order given in SAM/BAM header
      - based on order of sequences in FASTA used to build reference
  - ***sorting is very compute, I/O and memory intensive!***
    - can take hours for large BAMs
- Indexing a locus-sorted BAM allows fast random access
  - creates a small, binary alignment index file **(.bai)**
  - quite fast

Handling Duplicates



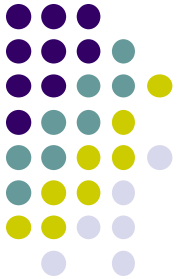
- Optional step, but very important for many protocols
- Definition of ***alignment duplicates***:
  - for single-end reads, or singleton/discordant paired-end reads:
    - alignments have the same ***start*** positions; actual sequence not considered
  - for properly paired reads:
    - pairs have same ***external*** coordinates (5' + 3' coordinates of the ***insert***)
    - actual insert sequence not considered
- Two choices for handling:
  - **samtools rmdup** – ***removes*** duplicates entirely
    - fast, but data is lost
    - does not intelligently handle data from multiple lanes
  - **Picard MarkDuplicates** – ***flags*** duplicates only (`0x400` BAM flag)
    - slower, but all alignments are retained
    - alignments from different lanes/replicates can be considered separately
  - both tools are quirky in their own ways

Alignment Workflow



http://samtools.sourceforge.net/samtools.shtml

Alignment metrics



- **samtools flagstat**
  - simple statistics based on alignment record flag values
    - total sequences (R1+R2); total mapped (`0x4 flag = 0`)
    - number properly paired (`0x2 flag = 1`)
    - number of duplicates (`0x400 flag = 1` if duplicates were marked)
  - BAM file must be indexed

```
161490318 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 secondary
0 + 0 supplementary
31602827 + 0 duplicates
158093331 + 0 mapped (97.90% : N/A)
161490318 + 0 paired in sequencing
80745159 + 0 read1
80745159 + 0 read2
153721151 + 0 properly paired (95.19% : N/A)
156184878 + 0 with itself and mate mapped
1908453 + 0 singletons (1.18% : N/A)
1061095 + 0 with mate mapped to a different chr
606632 + 0 with mate mapped to a different chr (mapQ>=5)
```

Alignment metrics

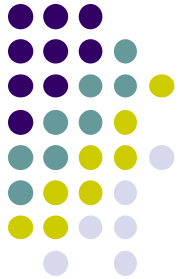
- **samtools idxstats**
  - reports number of reads aligning to each contig

contig	*length*	*# mapped*	*# not mapped*
chrI	230218	553609	2183
chrII	813184	1942996	5605
chrIII	316620	764449	2246
chrIV	1531933	3630237	10049
chrV	576874	1432940	4149
chrVI	270161	658338	1859
chrVII	1090940	2628838	7283
chrVIII	562643	1347702	4064
chrIX	439888	1079444	3057
chrX	745751	1861421	5576
chrXI	666816	1595615	4026
chrXII	1078177	4595061	23201
chrXIII	924431	2253102	6260
chrXIV	784333	1861773	5367
chrXV	1091291	2625205	7080
chrXVI	948066	2266237	6233
chrM	85779	210993	956
*	0	0	2291804

For alignments to *transcripts*

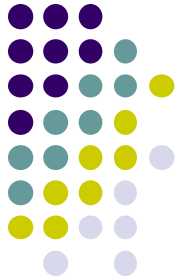
- contig names will be transcript names
- the # mapped is your initial quantification measure!

samtools notes



- There are 2 main “eras” of the **samtools** program
  - “old” **samtools**
    - v 0.1.19 last stable version
  - “new” **samtools**
    - v 1.0, 1.1, 1.2 – avoid these (very buggy!)
    - v 1.3+ stable
  - some functions have different arguments!
- **samtools** v 1.3+ has several new features
  - **samtools stats**
    - produces ***many*** different statistical reports
  - faster sorting
    - can use multiple threads

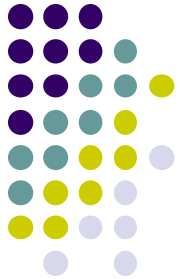
Computing average insert size



- Needed for some downstream analysis
  - e.g. ChIP-seq or RNA-seq alignment
- Simple **awk** script that computes average insert size for a BAM
  - **-F 0x4** filter to **samtools view** says only consider *mapped* reads
    - technically "***not unmapped***"
  - the **-f 0x2** filter says consider only ***properly paired*** reads
    - they have reliable "insert size" values in column 9
  - insert size values are negative for minus strand reads
    - can ignore because each proper pair will have one plus and one minus strand alignment, with same insert size

```
samtools view -F 0x4 -f 0x2 my_pe_data.bam | awk \
  'BEGIN{ FS="\t"; sum=0; nrec=0; }
   { if ($9 > 0) {sum += $9; nrec++;} }
   END{ print sum/nrec; }'
```

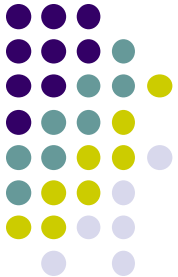
Interpreting alignment metrics



- Table below is taken from a spreadsheet I keep on Iyer lab alignments
  - all are yeast paired-end read datasets from ChIP-seq experiments
- Alignment rates
  - samples 1-3 have excellent alignment rates & good rates of proper pairing
  - sample 4
    - has an unusually low alignment rate for a ChIP-seq dataset
    - has a median insert size of only 109, and these were un-trimmed 50 bp reads
    - could 3' adapter contamination be affecting the alignment rate?
      - try re-aligning the sequences after trimming, say to 35 bases
      - see if the alignment rate improves

#	totSeq	totAlign	% align	numPair	pePrAln	% prPr	nDup	% dup	multiHit	% multi	iszMed
1	149,644,822	145,228,810	97.0%	74,822,411	72,221,545	96.5%	49,745,225	34%	16,216,807	11%	181
2	981,186	860,940	87.7%	490,593	424,915	86.6%	609,378	71%	127,987	15%	148
3	22,573,348	21,928,789	97.1%	11,286,674	10,783,971	95.5%	9,408,725	43%	3,711,004	17%	132
4	7,200,628	3,460,992	48.1%	3,600,314	1,626,121	45.2%	1,234,524	36%	649,690	19%	109

Interpreting alignment metrics

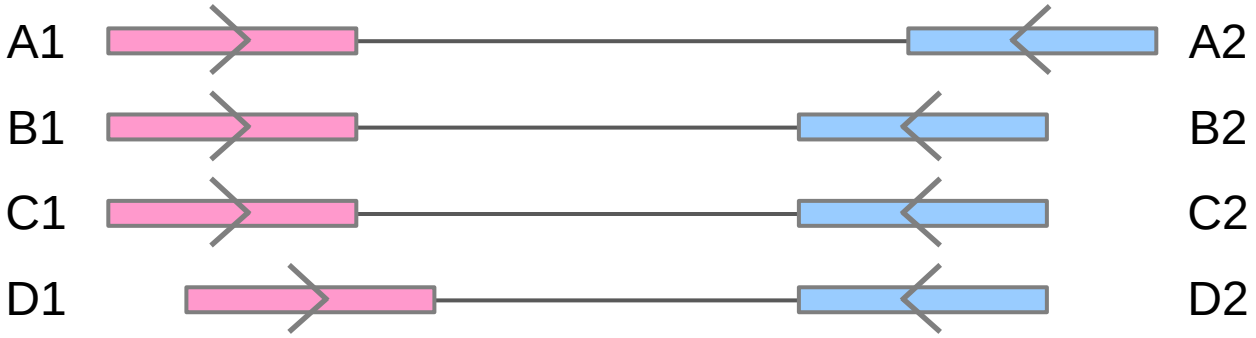


- Duplication rates
  - sample 2 is not very deeply sequenced but has a high duplication rate (71%)
    - subtracting duplicates from total aligned leaves only ~250,000 non-dup reads
      - not enough for further analysis (prefer 500,000+)
  - sample 3 has reasonable sequencing depth with substantial duplication (43%)
    - still leaves plenty of non-duplicate reads (> 12 million)
  - sample 1 is incredibly deeply sequenced
    - this is a control dataset (Mock ChIP), so is a great control to use (very complex!)
    - has a *very* low duplication rate (34%) considering that the yeast genome is only ~12 Mbase
      - ~145M mapped / 24M bases (+/- strands) should be ~6x coverage of every position!
      - so how is this low duplication rate possible?

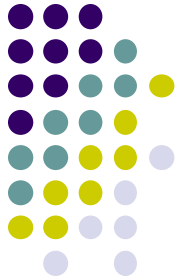
#	totSeq	totAlign	% align	numPair	pePrAln	% prPr	nDup	% dup	multiHit	% multi	iszMed
1	149,644,822	145,228,810	97.0%	74,822,411	72,221,545	96.5%	49,745,225	34%	16,216,807	11%	181
2	981,186	860,940	87.7%	490,593	424,915	86.6%	609,378	71%	127,987	15%	148
3	22,573,348	21,928,789	97.1%	11,286,674	10,783,971	95.5%	9,408,725	43%	3,711,004	17%	132
4	7,200,628	3,460,992	48.1%	3,600,314	1,626,121	45.2%	1,234,524	36%	649,690	19%	109

Read vs fragment duplication

- Consider the 4 fragments below
  - 4 R1 reads (pink), 4 R2 reads (blue)
- Duplication when only 1 end considered
  - A1, B1, C1 have identical sequences, D1 different
    - 2 unique + 2 duplicates = 50% duplication rate
  - B2, C2, D2 have identical sequences, A2 different
    - 2 unique + 2 duplicates = 50% duplication rate
- Duplication when both ends considered
  - fragments B and C are duplicates (same external sequences)
    - 3 unique + 1 duplicate = 25% duplication rate



Alignment wrap up



- Many tools involved
  - choose one or two and learn their options well
- Many steps are involved in the full alignment workflow
  - important to go through manually a few times for learning
    - but gets tedious quickly!
  - best practice
    - automate series of complex steps by wrapping into a ***pipeline script***
    - e.g. **bash** or **python** script
- Bioinformatics team has a set of pipeline scripts available
  - at TACC: in shared project directory **/work/projects/BioITeam/common/script/**
    - **align_bowtie2_illumina.sh**, **align_bwa_illumina.sh**, **trim_adapters.sh**, etc.
  - also available in **/mnt/bioi/script** on BRCF pods

Final thoughts

- Good judgement comes from experience

  unfortunately…
- Experience comes from bad judgement!
- So go get started making your 1st 1,000 mistakes….